JUEGOS DE ORDENADOR

ESPACIO

PARA SPECTRUM, ZX81, APPLE II, VIC 20, COMMODORE 64 Y MSX



EDICIONES GENERALES ANAYA

ESPACIO

Daniel Isaaman
y Jenny Tyler

Contenido

Ilustrado por
Martin Newton, Tony Baskeyfield, Graham Round, Jim Bamber
Mark Duffin y John Bolton

Diseñado por
Graham Round y Roger Priddy

Traducido por
Fernando García Fernández

El programa «Golpea los ojos de la oruga» ha sido escrito por Bob Merry
El programa «Despegue espacial» ha sido escrito por Richard Nash

Ediciones Generales Anaya

Acerca de este libro

Este libro contiene programas sencillos de juegos para usar con un microordenador ZX81, ZX Spectrum, MSX, VIC 20, Commodore 64 o Apple. Y algunos son lo suficientemente pequeños como para que quepan en la memoria de 1 K de un ZX81.

La mayoría de los micros usan el lenguaje BASIC, pero todos ellos tienen sus propias variaciones y dialectos. El listado que se da de cada programa en este libro corresponde al ZX81, y las líneas que tienen que cambiarse para otros ordenadores se marcan con un símbolo y se imprimen debajo. El hecho de que los programas hayan sido hechos para varias máquinas implica que no aprovechan todas las capacidades de cada una. Puedes intentar encontrar modos de hacer que los programas sean más cortos y bonitos.

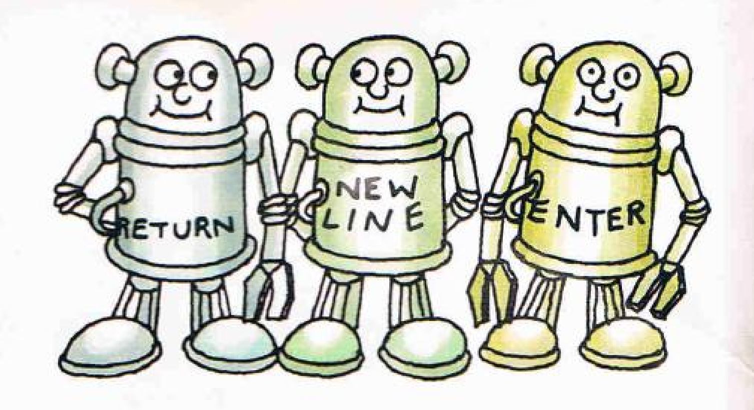
Para cada juego, se dan ideas para ampliar y modificar el programa y en la parte final del libro encontrarás ayudas y consejos para escribir tus propios juegos. También hallarás una tabla de conversión para ayudarte a adaptar los programas de revistas y otros libros a tu ordenador y un resumen de todos los términos del BASIC usados en este libro.

Tecleado de los programas

Las líneas que es necesario cambiar para otro ordenador que no sea el ZX81, se marcan con los siguientes símbolos:

▲ VIC 20 y Commodore 64
★ MSX
● Apple
■ ZX Spectrum
✸ ZX81

Cada vez que veas el símbolo del ordenador que estás usando, mira a la parte inferior de la hoja y teclea en su lugar la línea con el mismo número y con el mismo símbolo.



Recuerda estos puntos

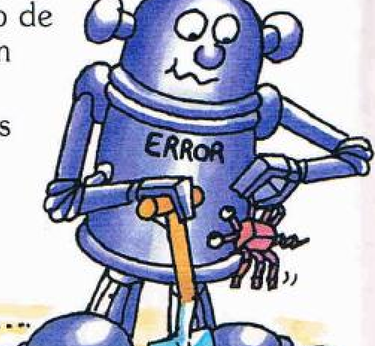
1. Teclea las líneas tal como están impresas, incluyendo todos los signos de puntuación y los espacios.
2. Teclea RETURN o ENTER al final de cada línea del listado.
3. Comprueba cada línea cuando la termines.
4. Asegúrate de que no te olvidas de ninguna línea ni confundes una con otra. Un trozo de papel o una regla es útil para señalar la línea por la que vas.
5. Mira los signos y asegúrate de que usas las líneas correctas para tu ordenador.
6. Si usas un ZX81 o un ZX Spectrum, recuerda que no tienes que teclear las instrucciones letra a letra, sino usar las teclas especiales existentes para cada instrucción.

Puede resultarte más fácil que alguien vaya leyendo el listado mientras lo tecleas. No te olvides de explicarle que debe leer cada coma, punto, paréntesis y espacio y que tiene que diferenciar entre la letra "O" y el cero y entre la "l" y el uno.

Eliminación de errores

Cuando hayas tecleado el programa, mira el manual para ver cómo puedes conseguir que salga en pantalla. (Normalmente es la instrucción LIST seguida por el número de las líneas de la sección que quieres ver.)

Comprueba que has tecleado todo





correctamente. Es fácil confundirse, de modo que no te asustes si aparece algún error. Mira tu manual para ver cómo puedes hacer cambios al programa una vez que lo has tecleado. Si no estás seguro, siempre puedes volver a teclear toda la línea. Todos los ordenadores sustituyen una línea ya existente por una nueva con el mismo número.

Aquí te damos una lista de los errores que más frecuentemente se suelen cometer:

1. Líneas sin teclear.
2. Líneas mal numeradas.
3. Dos líneas seguidas.
4. Se te puede haber olvidado algún signo de puntuación: paréntesis, comas, puntos o espacios; sobre todo en líneas complicadas y largas. Comprueba especialmente que existan tantos paréntesis abiertos como cerrados.
5. Puedes haber tecleado una línea que no corresponda a tu ordenador.
6. Confundir la letra "O" con un cero, o la "l" con un uno.
7. Usar números erróneos. Por ejemplo, incluyendo ceros de más.

Jugando con los programas

Para empezar a jugar, teclea RUN. En algunos juegos, las cosas suceden muy rápidamente, por lo que tienes que leerte las instrucciones antes para saber qué es es lo que tienes que hacer.

Es muy probable que el programa siga teniendo algún error y que no funcione o funcione mal. A veces el ordenador se parará y te dará un error. Para saber qué significa puedes mirar el manual. Éste te puede ayudar a encontrar el error, pero no siempre lo hará. Lista el programa de nuevo y compáralo con el del libro con mucha atención.

Cuando el programa se termine, el ordenador imprimirá una frase del tipo "BREAK IN LINE 200". Para jugar de nuevo, teclea RUN.

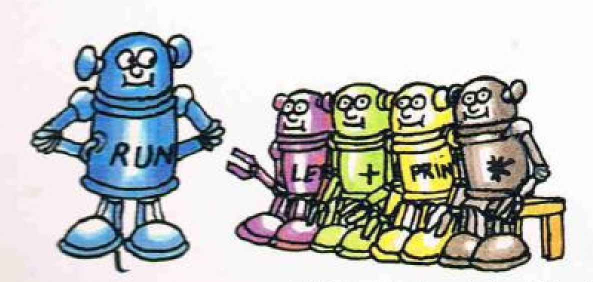
Experimenta con los juegos

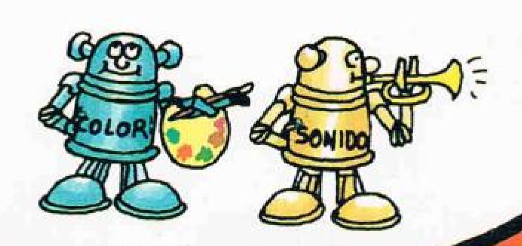
A lo largo del libro hay sugerencias para modificar y hacer añadidos a los programas, pero no tengas miedo de hacer otros que se te ocurran. No estropearás el ordenador y siempre puedes volver al original si lo que has cambiado no funciona.

Probablemente querrás ajustar la velocidad de algunos juegos*, especialmente cuando ya tengas experiencia con ellos. Para ello, se indica en la página de cada programa qué línea hay que cambiar.

En los sitios en los que aparece la instrucción PRINT, puedes cambiar el mensaje que hay entre las comillas por otro cualquiera que tú prefieras. También, a menos que tengas un ZX81 con sólo 1 K de memoria, podrás añadir más mensajes. Para hacerlo, teclea un número de línea (por ejemplo 105 si quieres ponerlo entre la 100 y la 110) seguido de PRINT y del mensaje entre comillas.

Si tu ordenador tiene colores y sonido, puedes mejorar tu juego añadiéndoselos al programa, siguiendo las instrucciones que se den en el manual del ordenador.





* *Si tienes un Spectrum o un MSX mira la página 37, donde se te hacen indicaciones.*

Despegue espacial

Eres el capitán de una nave espacial y te has estrellado con tu nave en un planeta desconocido del que debes salir rápidamente en otra nave que has capturado. El ordenador de ésta te indica la gravedad del planeta. Tú tienes que averiguar la fuerza necesaria para poder despegar. Si la que das es muy baja, la nave no despegará, y si es muy alta, el mecanismo de seguridad entrará en acción impidiendo que se destruya la nave espacial. Si todavía sigues en el planeta después de 10 intentos, sus habitantes te capturarán.



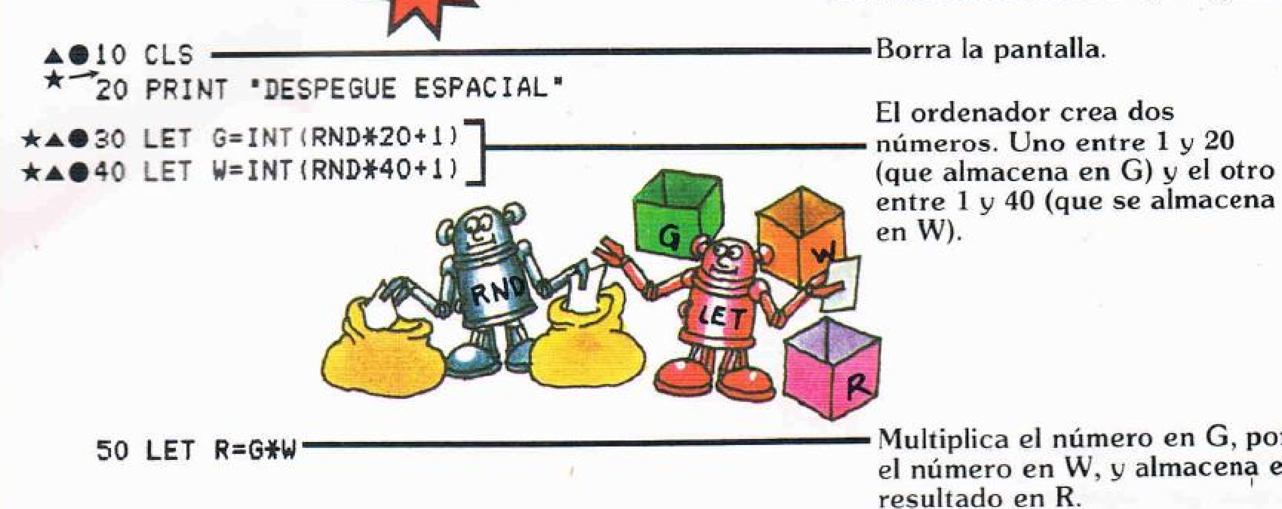


Cómo funciona el programa

```
▲●10 CLS
★→20 PRINT "DESPEGUE ESPACIAL"
★▲●30 LET G=INT(RND*20+1)
★▲●40 LET W=INT(RND*40+1)
```

Borra la pantalla.

El ordenador crea dos números. Uno entre 1 y 20 (que almacena en G) y el otro entre 1 y 40 (que se almacena en W).




```
50 LET R=G*W
```

Multiplica el número en G, por el número en W, y almacena el resultado en R.

```
60 PRINT "GRAVEDAD= ";G
```

Imprime GRAVEDAD y el número en G.

```
70 PRINT "TECLEA LA FUERZA"
```

Te pide un número.

```
80 FOR C=1 TO 10
```

Esto empieza un bucle que le dice al ordenador que repita la siguiente sección 10 veces, para darte diez intentos.

```
90 INPUT F
```

Guarda tu número en F.

```
100 IF F>R THEN PRINT "DEMASIADO FUERTE";
110 IF F<R THEN PRINT "DEMASIADO FLOJO";
120 IF F=R THEN GOTO 190
```

Compara el número en F con el número en R, e imprime el mensaje adecuado o salta a la línea 190.

```
130 IF C<>10 THEN PRINT ", INTENTALO DE NUEVO"
```

Lo imprime si lo has intentado menos de diez veces y no has hallado la respuesta correcta.

```
140 NEXT C
```

Fin del bucle. Vuelve a la línea 80 para otro turno.

```
150 PRINT
160 PRINT "HAS FALLADO -"
170 PRINT "TE HAN COGIDO"
180 STOP
190 PRINT "BUEN DESPEGUE"
```

Esto lo imprime después de diez intentos fallidos.

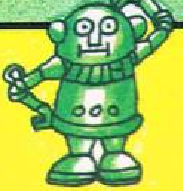


Este listado funcionará en un ZX81.
Para otros ordenadores, haz estos cambios:

```
●10 HOME
▲10 PRINT CHR$(147)
  {15 PRINT "INTRODUCE UN NUMERO POSITIVO"
★ {17 INPUT Z
  {18 Z=RND(-Z)
★▲●30 LET G=INT(RND(1)*20)
★▲●40 LET W=INT(RND(1)*40)
```

Cómo complicar el juego

Puedes cambiar el programa para que te deje menos de 10 intentos. Esto se hace alterando el último número de la línea 80 y el de la línea 130. (Los dos tienen que ser iguales.)



Rincón de los problemas

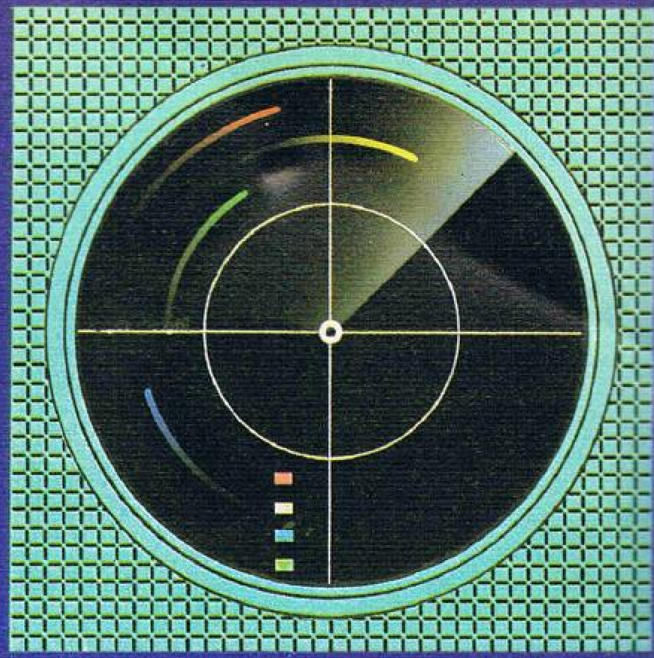
Puedes cambiar los límites entre los que se encuentran las fuerzas. ¡A ver si averiguas cómo!

▲★●■ *mira la página 2.*

Juegos intergalácticos

Hay una feroz lucha entre las compañías de televisión de todo el mundo para conseguir la exclusiva de los Primeros Juegos Intergalácticos. Todo depende de qué compañía consiga poner primero su satélite en la órbita correcta.

Tú eres el ingeniero encargado de lanzar el de la compañía de televisión Nuevo Siglo. Las decisiones cruciales acerca del ángulo y la velocidad de lanzamiento del cohete las tienes que tomar tú. ¿Podrás?



Cómo funciona el programa

```
★→10 PRINT "JUEGOS INTERGALACTICOS"
★▲●20 LET H=INT(RND*100+1)
   30 PRINT "DEBES LANZAR UN SATELITE"
   40 PRINT "A UNA ALTURA DE ";H
```
Elige la altura a la que debes lanzar tu satélite, la mete en H y la imprime.

```
   50 FOR G=1 TO 8
```
Principio de un bucle que te permite realizar 8 intentos.

```
   60 PRINT "INTRODUCE ANGULO (0-90)"
   70 INPUT A
```
Te pide un ángulo y lo almacena en A.

```
   80 PRINT "INTRODUCE VELOCIDAD (0-40000)"
   90 INPUT V
```
Te pide una velocidad y la almacena en V.

```
  100 LET A=A-ATN(H/3)*180/3.14159
```
Usa H para calcular el ángulo correcto y se lo resta al que tú has tecleado para ver cuánto te has acercado.

```
  110 LET V=V-3000*SQR(H+1/H)
```
Calcula la velocidad correcta y se la resta a la que has dado.

```
  120 IF ABS(A)<2 AND ABS(V)<100 THEN GOTO 210
```
Comprueba si estuviste lo suficientemente cerca como para acertar y, si es así, salta a la 210.

```
  130 IF A<-2 THEN PRINT "DEMASIADO INCLINADO"
  140 IF A>2 THEN PRINT "DEMASIADO VERTICAL"
  150 IF V<-100 THEN PRINT "DEMASIADO LENTO"
  160 IF V>100 THEN PRINT "DEMASIADO RAPIDO"
```
Imprime el comentario apropiado para ayudarte en tu próximo intento.

```
  170 NEXT G
```
Vuelve a hacer otro intento.

```
  180 PRINT "HAS FALLADO"
  190 PRINT "ESTAS DESPEDIDO"
  200 STOP
```
Imprime esto después de 8 intentos equivocados.

```
  210 PRINT "LO HAS LOGRADO"
  220 PRINT "NCTV GANA-GRACIAS A TI"
  230 STOP
```
Imprime esto si acertaste.

Este listado funcionará en un ZX81. Para otros ordenadores, haz estos cambios:

```
  ★{15 PRINT "INTRODUCE UN NUMERO POSITIVO"
    17 INPUT Z
    18 Z=RND(-Z)
★▲●20 LET H=INT(RND(1)*100+1)
```

Ampliando el programa

Estas tres líneas extras harán que el ordenador te dé más o menos puntos, según la rapidez con que hayas despegado correctamente.

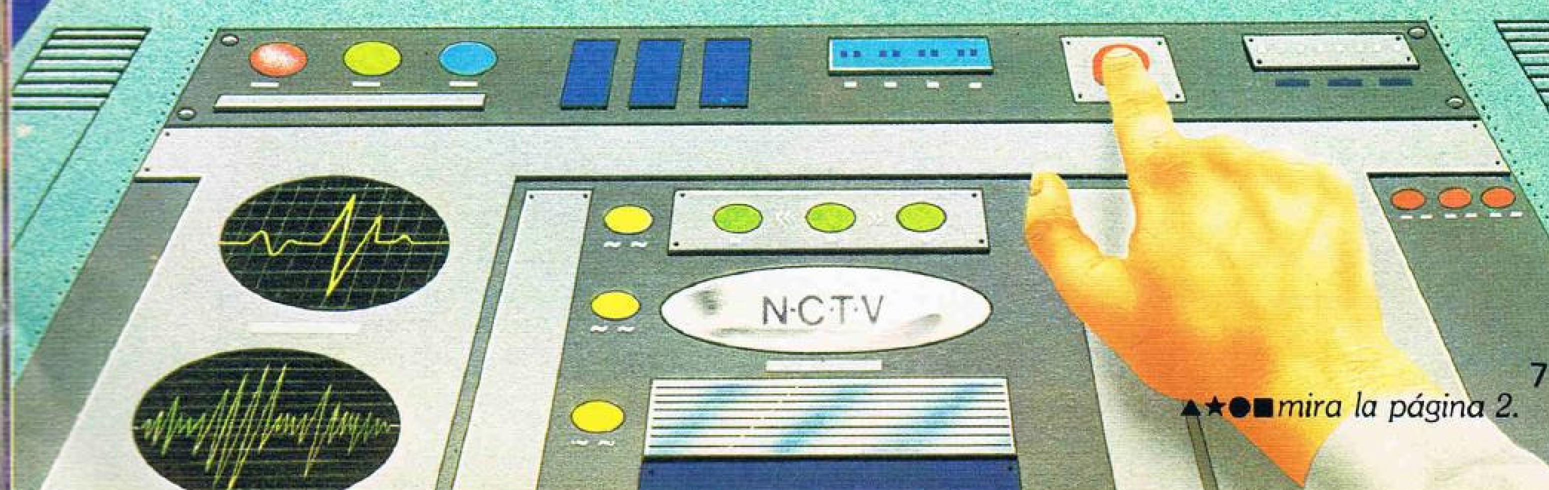
```
222 LET B=INT(1000/G)
225 PRINT "HAS GANADO UNA "
227 PRINT "BONIFICACION DE ";B;
    " CREDITOS"
```



Rincón de los problemas

¿Puedes cambiar el programa de modo que, si ganas, vuelva a empezar automáticamente añadiendo los puntos que ganes a todos los que hubieses conseguido anteriormente? (Necesitas cambiar dos líneas y añadir una.)

Mira a ver cuánto tiempo puedes jugar antes de que la TVNS te despida.

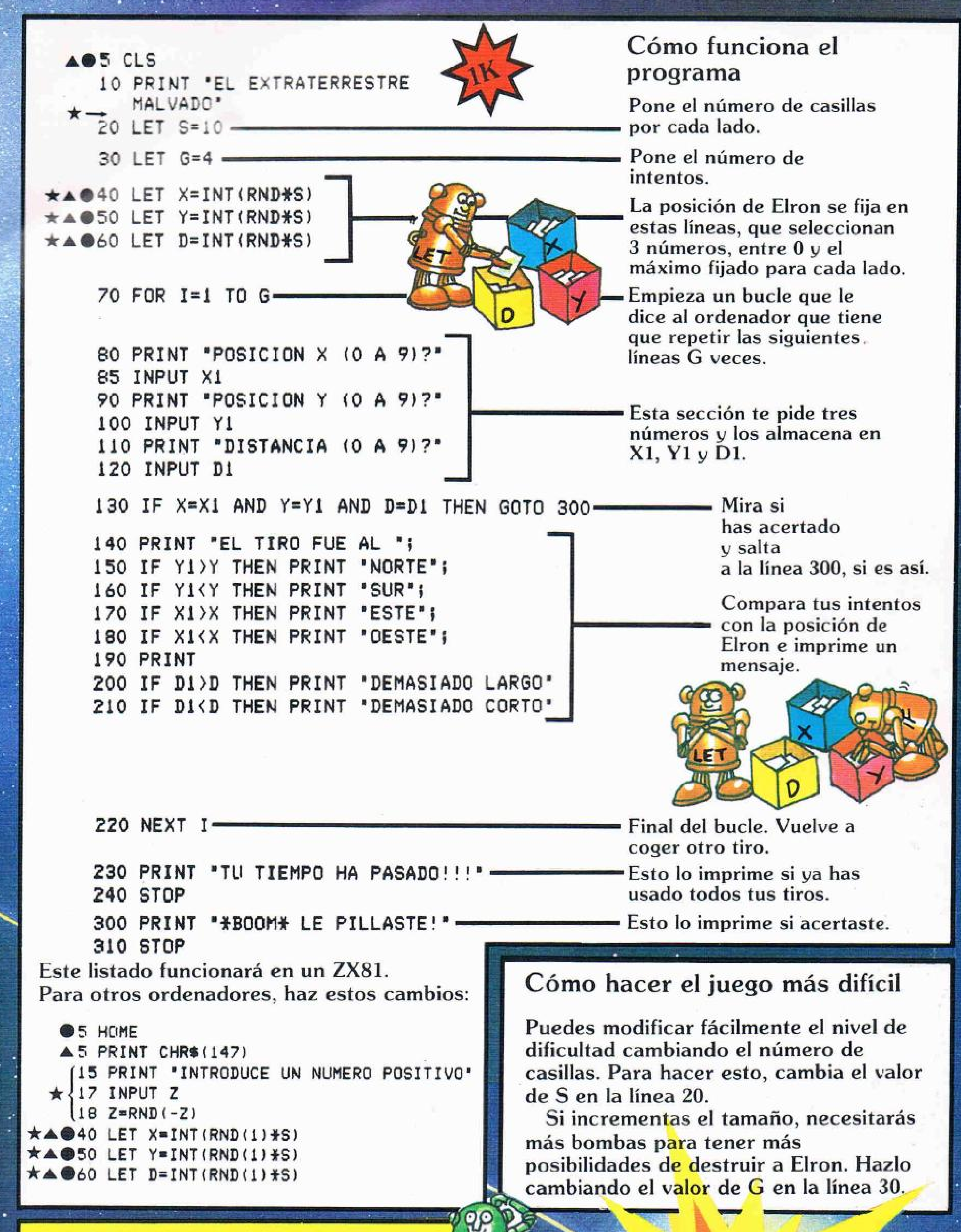


▲★●■ *mira la página 2.*

El malvado extraterrestre

En algún lugar cerca de ti, en el espacio profundo y oscuro, habita Elron, el Malvado. Has conseguido desactivar todas sus armas, excepto las de corto alcance, y todavía puede volver invisible su nave. Tú sabes que está en alguna parte del tablero tridimensional que puedes ver en la pantalla de tu ordenador (mira la figura de abajo); pero, ¿dónde? Tienes cuatro bombas espaciales. Cada una puede estallar en una casilla que le indiques con tres números, entre 0 y 9, que te pedirá el ordenador. ¿Puedes echar al malvado Elron del espacio antes de que se deslice hacia ti y te destruya?





Cómo funciona el programa

```
▲●5 CLS
  10 PRINT "EL EXTRATERRESTRE
     MALVADO"
★→20 LET S=10
```

Pone el número de casillas por cada lado.

```
  30 LET G=4
```

Pone el número de intentos.

```
★▲●40 LET X=INT(RND*S)
★▲●50 LET Y=INT(RND*S)
★▲●60 LET D=INT(RND*S)
```

La posición de Elron se fija en estas líneas, que seleccionan 3 números, entre 0 y el máximo fijado para cada lado.

```
  70 FOR I=1 TO G
```

Empieza un bucle que le dice al ordenador que tiene que repetir las siguientes líneas G veces.

```
  80 PRINT "POSICION X (0 A 9)?"
  85 INPUT X1
  90 PRINT "POSICION Y (0 A 9)?"
 100 INPUT Y1
 110 PRINT "DISTANCIA (0 A 9)?"
 120 INPUT D1
```

Esta sección te pide tres números y los almacena en X1, Y1 y D1.

```
 130 IF X=X1 AND Y=Y1 AND D=D1 THEN GOTO 300
```

Mira si has acertado y salta a la línea 300, si es así.

```
 140 PRINT "EL TIRO FUE AL ";
 150 IF Y1>Y THEN PRINT "NORTE";
 160 IF Y1<Y THEN PRINT "SUR";
 170 IF X1>X THEN PRINT "ESTE";
 180 IF X1<X THEN PRINT "OESTE";
 190 PRINT
 200 IF D1>D THEN PRINT "DEMASIADO LARGO"
 210 IF D1<D THEN PRINT "DEMASIADO CORTO"
```

Compara tus intentos con la posición de Elron e imprime un mensaje.

```
 220 NEXT I
```

Final del bucle. Vuelve a coger otro tiro.

```
 230 PRINT "TU TIEMPO HA PASADO!!!"
 240 STOP
```

Esto lo imprime si ya has usado todos tus tiros.

```
 300 PRINT "*BOOM* LE PILLASTE!"
 310 STOP
```

Esto lo imprime si acertaste.

Este listado funcionará en un ZX81.
Para otros ordenadores, haz estos cambios:

```
  ●5 HOME
  ▲5 PRINT CHR$(147)
   ⎧15 PRINT "INTRODUCE UN NUMERO POSITIVO"
  ★⎨17 INPUT Z
   ⎩18 Z=RND(-Z)
★▲●40 LET X=INT(RND(1)*S)
★▲●50 LET Y=INT(RND(1)*S)
★▲●60 LET D=INT(RND(1)*S)
```

Cómo hacer el juego más difícil

Puedes modificar fácilmente el nivel de dificultad cambiando el número de casillas. Para hacer esto, cambia el valor de S en la línea 20.

Si incrementas el tamaño, necesitarás más bombas para tener más posibilidades de destruir a Elron. Hazlo cambiando el valor de G en la línea 30.

Rincón de los problemas

¿Puedes modificar el programa de modo que el ordenador te pida un número de dificultad que pueda almacenar en S en vez de dejarlo fijo? (Ayuda: limita el valor de S a uno entre 6 y 30 y usa INT(S/3) para el valor de G en la línea 30.)

▲★●■ mira la página 2.



Golpea los ojos de la oruga

¡Estás atrapado! En cualquier lugar al que miras, encuentras la fría mirada de una oruga espacial que se esconde rápidamente detrás de una roca. Se dirige lentamente hacia ti, cercándote y esperando el momento propicio para agarrarte con sus antenas pegajosas. Afortunadamente, tienes tu pistola de protones.

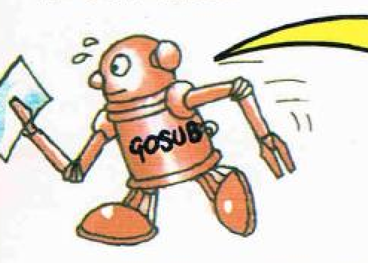
Las orugas aparecen en cuatro sitios diferentes de la pantalla, que corresponden respectivamente a las teclas 1 a 4. Pulsa la correcta mientras los ojos están en la pantalla y las destruirás. Hay diez orugas en total.

Cómo funciona el programa

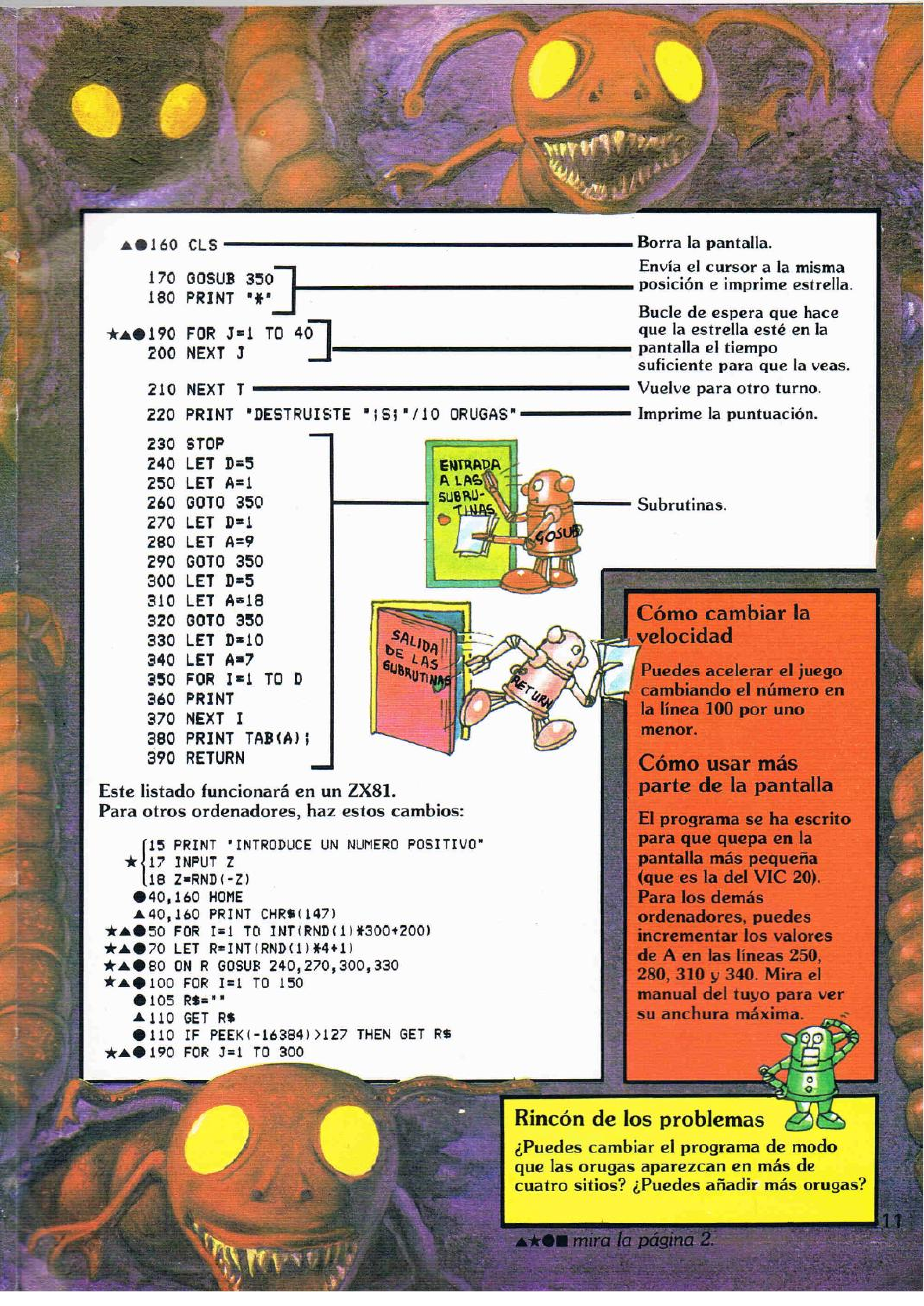
Programa	Explicación
`10 PRINT "OJOS DE ORUGA"` ★→ `20 LET S=0`	Pone la puntuación a cero al principio del juego.
`30 FOR T=1 TO 10`	Principio de un bucle que te da 10 turnos.
▲● `40 CLS`	Borra la pantalla.
★▲● `50 FOR I=1 TO INT(RND*30+20)` `60 NEXT I`	Este bucle espera un tiempo variable, dependiendo del valor que genere RND.
★▲● `70 LET R=INT(RND*4+1)`	Elige un número de 1 a 4 y lo guarda en R.
★▲● `80 GOSUB 210+30*R`	Salta a una rutina de entre cuatro existentes, según el valor de R. Éstas obtienen dos números que corresponden a una posición de la pantalla —"A" espacios y "D" líneas— y salta de nuevo a 350 para mover el cursor a esta posición.
`90 PRINT "OO"`	Imprime los ojos de la oruga en esta posición.

★▲● `100 FOR I=1 TO 20`
●→▲● `110 LET R$=INKEY$`
`120 IF R$<>"" THEN GOTO 140`
`130 NEXT I`
`140 IF VAL("0"+R$)<>R THEN GOTO 210`
	Hace un bucle para ver si estás pulsando una tecla. Si lo estás haciendo salta a la línea 140 y mira si es la correcta.

`150 LET S=S+1`	Incrementa la puntuación en 1.



GOSUB hace que el ordenador salte desde la parte principal del programa a una 'subrutina" (en la página siguiente). El RETURN al final de la subrutina lo envía a la parte principal de nuevo.




```
▲●160 CLS                                    Borra la pantalla.

  170 GOSUB 350                              Envía el cursor a la misma
  180 PRINT "*"                              posición e imprime estrella.

★▲●190 FOR J=1 TO 40                         Bucle de espera que hace
  200 NEXT J                                 que la estrella esté en la
                                             pantalla el tiempo
                                             suficiente para que la veas.

  210 NEXT T                                 Vuelve para otro turno.

  220 PRINT "DESTRUISTE ";S;"/10 ORUGAS"     Imprime la puntuación.

  230 STOP                                   Subrutinas.
  240 LET D=5
  250 LET A=1
  260 GOTO 350
  270 LET D=1
  280 LET A=9
  290 GOTO 350
  300 LET D=5
  310 LET A=18
  320 GOTO 350
  330 LET D=10
  340 LET A=7
  350 FOR I=1 TO D
  360 PRINT
  370 NEXT I
  380 PRINT TAB(A);
  390 RETURN
```

Este listado funcionará en un ZX81. Para otros ordenadores, haz estos cambios:

```
    ⎧15 PRINT "INTRODUCE UN NUMERO POSITIVO"
  ★ ⎨17 INPUT Z
    ⎩18 Z=RND(-Z)
   ●40,160 HOME
   ▲40,160 PRINT CHR$(147)
 ★▲●50 FOR I=1 TO INT(RND(1)*300+200)
 ★▲●70 LET R=INT(RND(1)*4+1)
 ★▲●80 ON R GOSUB 240,270,300,330
 ★▲●100 FOR I=1 TO 150
   ●105 R$=""
   ▲110 GET R$
   ●110 IF PEEK(-16384)>127 THEN GET R$
 ★▲●190 FOR J=1 TO 300
```

Cómo cambiar la velocidad

Puedes acelerar el juego cambiando el número en la línea 100 por uno menor.

Cómo usar más parte de la pantalla

El programa se ha escrito para que quepa en la pantalla más pequeña (que es la del VIC 20). Para los demás ordenadores, puedes incrementar los valores de A en las líneas 250, 280, 310 y 340. Mira el manual del tuyo para ver su anchura máxima.

Rincón de los problemas

¿Puedes cambiar el programa de modo que las orugas aparezcan en más de cuatro sitios? ¿Puedes añadir más orugas?

▲★●■ *mira la página 2.*

Alunizaje

Estás al mando de un módulo lunar encargado de llevar a un equipo de astronautas a la Luna. Para poder llegar a salvo, tienes que descender lentamente, pero eso consume combustible y no tienes mucho.

Tu ordenador te dirá la altura y velocidad inicial, así como el combustible disponible y después te preguntará cuánto quieres usar. Cuando se lo digas, calculará tu nueva posición y velocidad. Un encendido de 5 mantendrá tu velocidad constante. Un número mayor, la reducirá. Intenta conseguir una velocidad tan próxima a cero como sea posible cuando alunices.

Fíjate en las comas y en los puntos y comas en las líneas 70 y 80. Prueba a quitarlos y a cambiarlos entre sí a ver qué sucede.

```
▲●10 CLS
20 PRINT "ALUNIZAJE"
30 LET T=0
40 LET H=500
50 LET V=50
60 LET F=120
70 PRINT "TIEMPO ";T;"ALTURA ";H
80 PRINT "VEL.";V,"FUEL";F
90 IF F=0 THEN GOTO 140
100 PRINT "ENCENDIDO? (0-30)"
110 INPUT B
120 IF B<0 THEN LET B=0
130 IF B>30 THEN LET B=30
140 IF B>F THEN LET B=F
150 LET V1=V-B+5
160 LET F=F-B
170 IF (V1+V)/2>=H THEN GOTO 220
180 LET H=H-(V1+V)/2
190 LET T=T+1
200 LET V=V1
210 GOTO 70
220 LET V1=V+(5-B)*H/V
230 IF V1>5 THEN PRINT "TE ESTRELLASTE-TODOS MUERTOS"
240 IF V1>1 AND V1<=5 THEN PRINT "OK- PERO ALGUNOS HERIDOS"
250 IF V1<=1 THEN PRINT "BUEN ATERRIZAJE."
260 STOP
```

Este listado funcionará en un ZX81.
Para otros ordenadores, haz estos cambios:

```
●10 HOME
▲10 PRINT CHR$(147)
```

 ▲★●■ *mira la página 2.*



Cómo funciona el programa

Pone los valores iniciales de tiempo, altura, velocidad y gasolina y los imprime.

Si no te queda gasolina, el ordenador se va a la parte final del programa, saltándose la sección que te pide el combustible a usar, e imprime un comentario de tu cercanía a la superficie.

Lee el número de gasolina que quieres usar y comprueba si está dentro de los límites correctos.

Calcula tu nueva velocidad, V1.

Calcula tu nuevo nivel de combustible.

Comprueba si la distancia recorrida en el último intento es mayor o igual que la altura sobre la Luna. Si es así, has aterrizado. El ordenador se va entonces al final del programa para ver qué tal alunizaje has hecho.

Calcula tu nueva altura.

Incrementa el tiempo en 1.

Pone tu nueva velocidad en V, de modo que la imprima en la línea 80 en tu próximo intento.

Vuelve al principio del bucle para un nuevo intento.

Calcula tu velocidad al hacer contacto y mira qué tal lo has hecho.

Ampliación del programa

Si añades las siguientes líneas, verás que se imprime una estrella en cada intento. La distancia de la estrella al lado izquierdo de la pantalla corresponde a tu altura sobre la Luna.

```
85 FOR I=2 TO H/500*nn
86 PRINT " ";
87 NEXT I
88 PRINT "*"
```

Reemplaza nn por el ancho de tu pantalla.

Cambios que puedes probar

Intenta cambiar los valores de H, V y F en las líneas 40 y 60 y mira qué sucede.

Rincón de los problemas

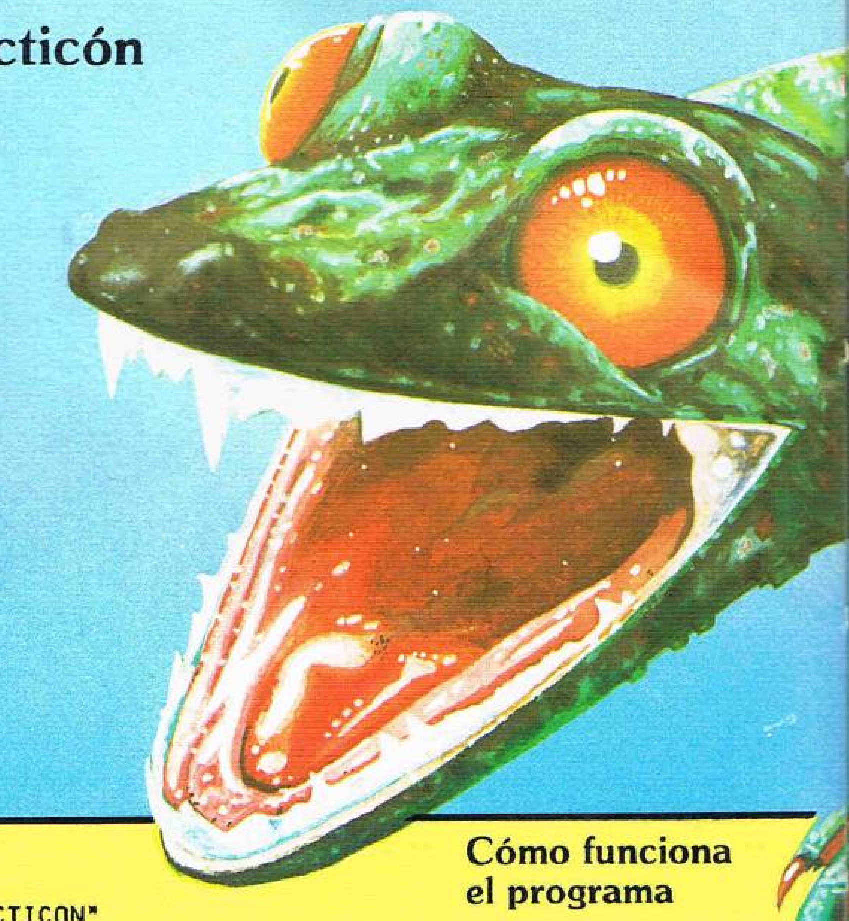
Puedes hacer el juego más sencillo incrementando la velocidad máxima permitida para un aterrizaje seguro. ¿Cómo cambiarías el programa para que hiciese esto?



Los monstruos de Galacticón

El aterrizaje en Galacticón fue fácil, pero nadie te avisó de que aquí se encontraban algunos de los monstruos más desagradables de todo el universo.

Cuando un monstruo te ataque, tendrás que elegir cuál de tus armas usar: Una pistola de rayos, un bazooka Trypton o una espada sónica. ¿Puedes hacer la elección correcta? Si es así, podrás vivir lo suficiente para conquistar Galacticón.



```
      10 PRINT "LOS MONSTRUOS DE GALACTICON"
  ★→
      20 DIM M$(4)
      30 LET N=4
      40 LET M=5
      50 LET M$(1)="SULFACIDOR"
      60 LET M$(2)="FLAMGONDAR"
      70 LET M$(3)="BALNOLOTIN"
      80 LET M$(4)="GOLANDOR"
      90 FOR I=1 TO N
 ★▲●100 LET A=INT(RND*N+1)
 ★▲●110 LET B=INT(RND*N+1)
     120 LET T$=M$(A)
     130 LET M$(A)=M$(B)
     140 LET M$(B)=T$
     150 NEXT I
     160 FOR T=1 TO 8
   ▲●170 CLS
 ★▲●180 LET R=INT(RND*N+1)
     190 PRINT "VIENE UN MONSTRUO..."
     200 PRINT "ES UN ";M$(R)
     210 PRINT "QUE ARMA ? (R,S,T)"
     220 INPUT R$
★▲●■230 LET W=CODE(R$)-54+R
    ▲240 LET W=W-3*(W>3)-3*(W>6)
     250 IF W=2 THEN GOTO 300
```

Puedes utilizar esta rutina de barajado siempre que quieras mezclar algo.

Cómo funciona el programa

Crea un sitio para almacenar datos (una "matriz") con el nombre M$, que tiene cuatro compartimentos, M$(1) a M$(4), para cada monstruo.

Hace que el número de monstruos sea 4.

Hace que el número de personas en tu grupo sea 5.

Pone los nombres de los monstruos en la matriz.

Estas líneas barajan los nombres de los monstruos. El ordenador ejecuta un bucle N veces. Cada vez selecciona dos números entre 1 y N y cambia entre sí los nombres de sus compartimentos. T$ es una cadena de almacenamiento temporal usada durante el cambio.

Empieza el bucle de 8 turnos.

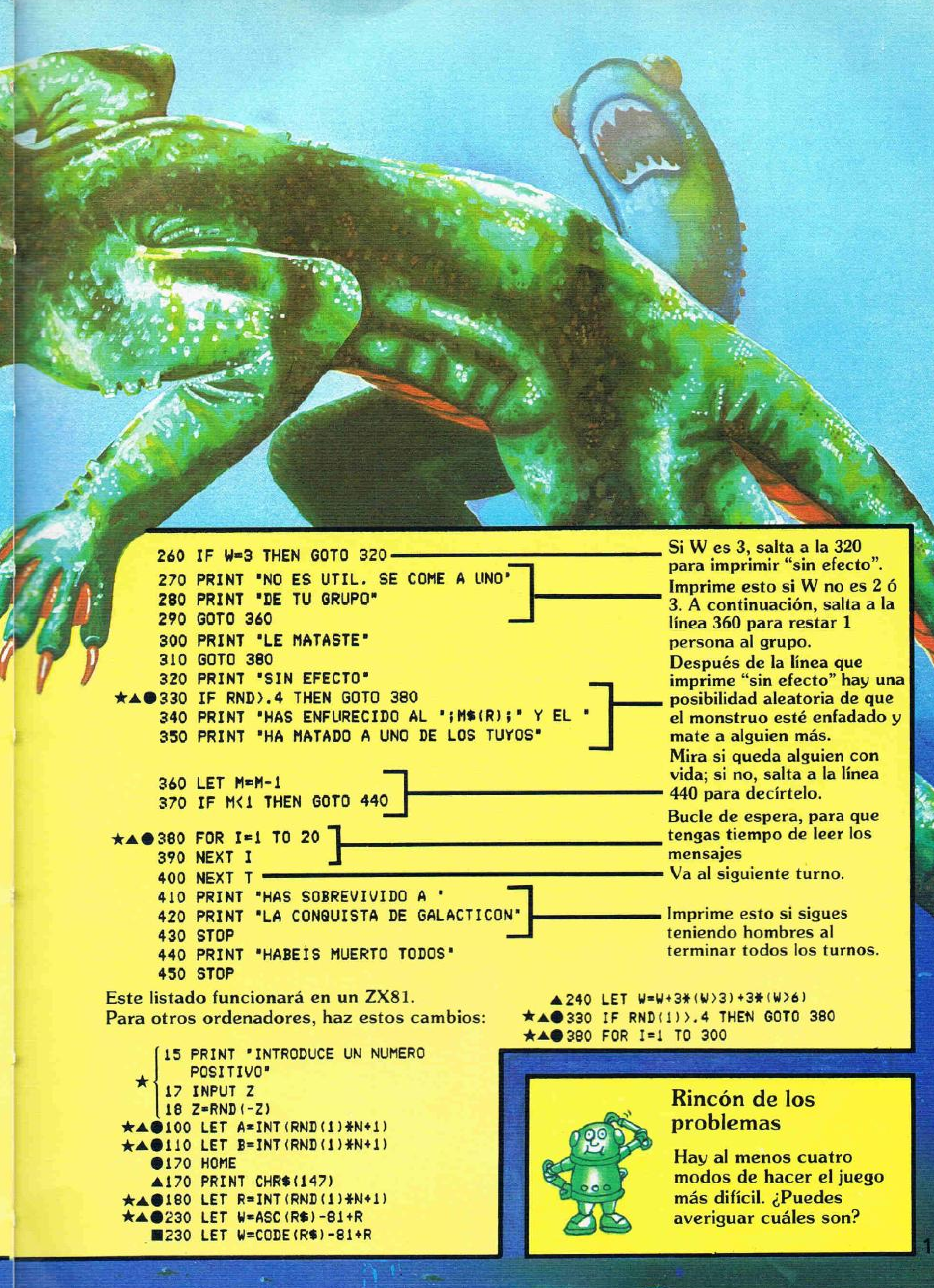
Borra la pantalla.

Elige uno de los monstruos e imprime su nombre.

El ordenador usa los valores en R y R$ para elegir un número de arma, W, que puede ser 1, 2 ó 3.

Si W es 2, el ordenador salta a la línea 300 para decirte que has matado al monstruo.

▲★●■ *mira la página 2.*



```
260 IF W=3 THEN GOTO 320
270 PRINT "NO ES UTIL. SE COME A UNO"
280 PRINT "DE TU GRUPO"
290 GOTO 360
300 PRINT "LE MATASTE"
310 GOTO 380
320 PRINT "SIN EFECTO"
★▲●330 IF RND>.4 THEN GOTO 380
340 PRINT "HAS ENFURECIDO AL ";M$(R);" Y EL "
350 PRINT "HA MATADO A UNO DE LOS TUYOS"
360 LET M=M-1
370 IF M<1 THEN GOTO 440
★▲●380 FOR I=1 TO 20
390 NEXT I
400 NEXT T
410 PRINT "HAS SOBREVIVIDO A "
420 PRINT "LA CONQUISTA DE GALACTICON"
430 STOP
440 PRINT "HABEIS MUERTO TODOS"
450 STOP
```

Si W es 3, salta a la 320 para imprimir "sin efecto".

Imprime esto si W no es 2 ó 3. A continuación, salta a la línea 360 para restar 1 persona al grupo.

Después de la línea que imprime "sin efecto" hay una posibilidad aleatoria de que el monstruo esté enfadado y mate a alguien más.

Mira si queda alguien con vida; si no, salta a la línea 440 para decírtelo.

Bucle de espera, para que tengas tiempo de leer los mensajes

Va al siguiente turno.

Imprime esto si sigues teniendo hombres al terminar todos los turnos.

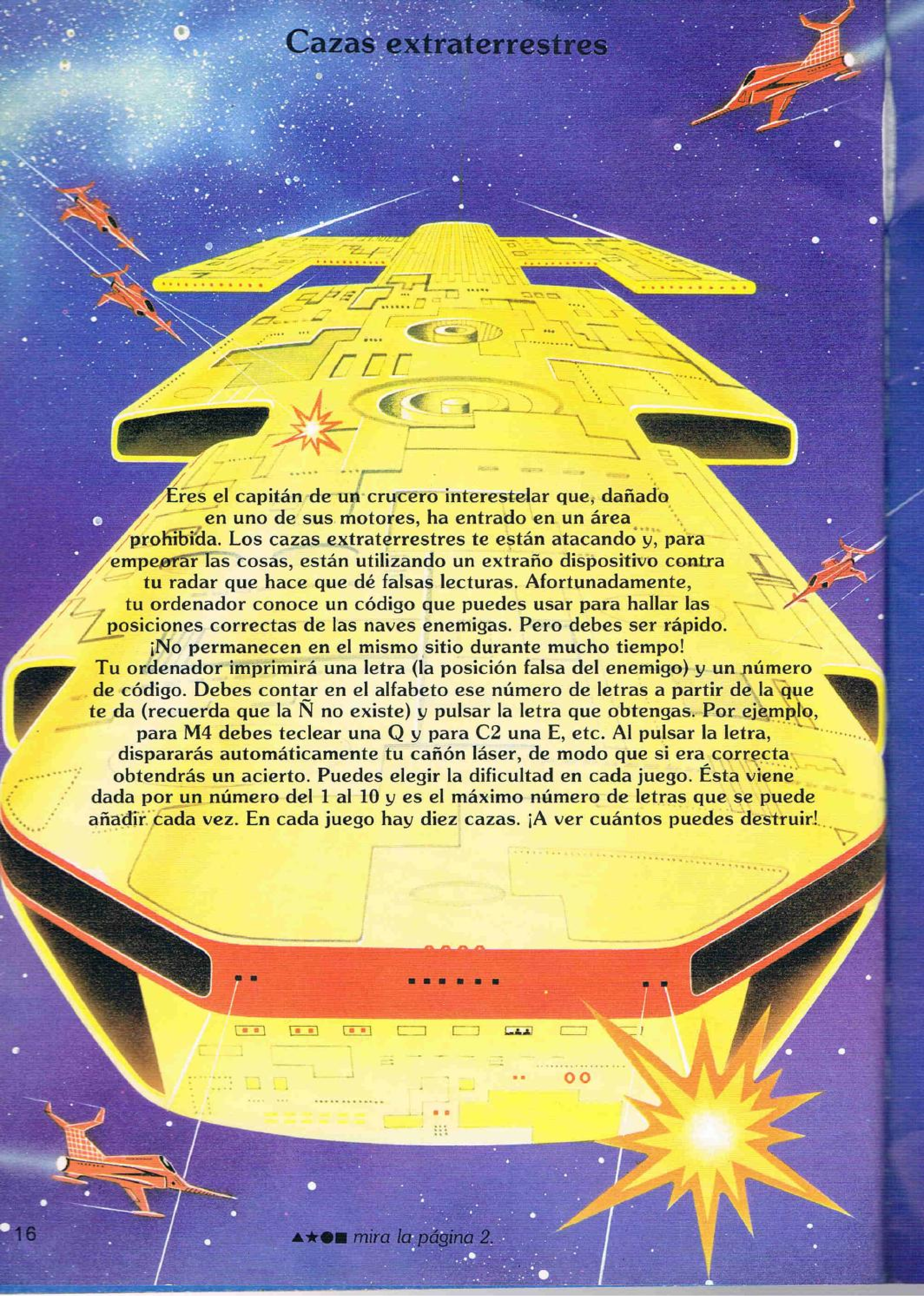
Este listado funcionará en un ZX81. Para otros ordenadores, haz estos cambios:

```
★ { 15 PRINT "INTRODUCE UN NUMERO
        POSITIVO"
    17 INPUT Z
    18 Z=RND(-Z)
★▲●100 LET A=INT(RND(1)*N+1)
★▲●110 LET B=INT(RND(1)*N+1)
●170 HOME
▲170 PRINT CHR$(147)
★▲●180 LET R=INT(RND(1)*N+1)
★▲●230 LET W=ASC(R$)-81+R
■230 LET W=CODE(R$)-81+R
▲240 LET W=W+3*(W>3)+3*(W>6)
★▲●330 IF RND(1)>.4 THEN GOTO 380
★▲●380 FOR I=1 TO 300
```

Rincón de los problemas

Hay al menos cuatro modos de hacer el juego más difícil. ¿Puedes averiguar cuáles son?

Cazas extraterrestres

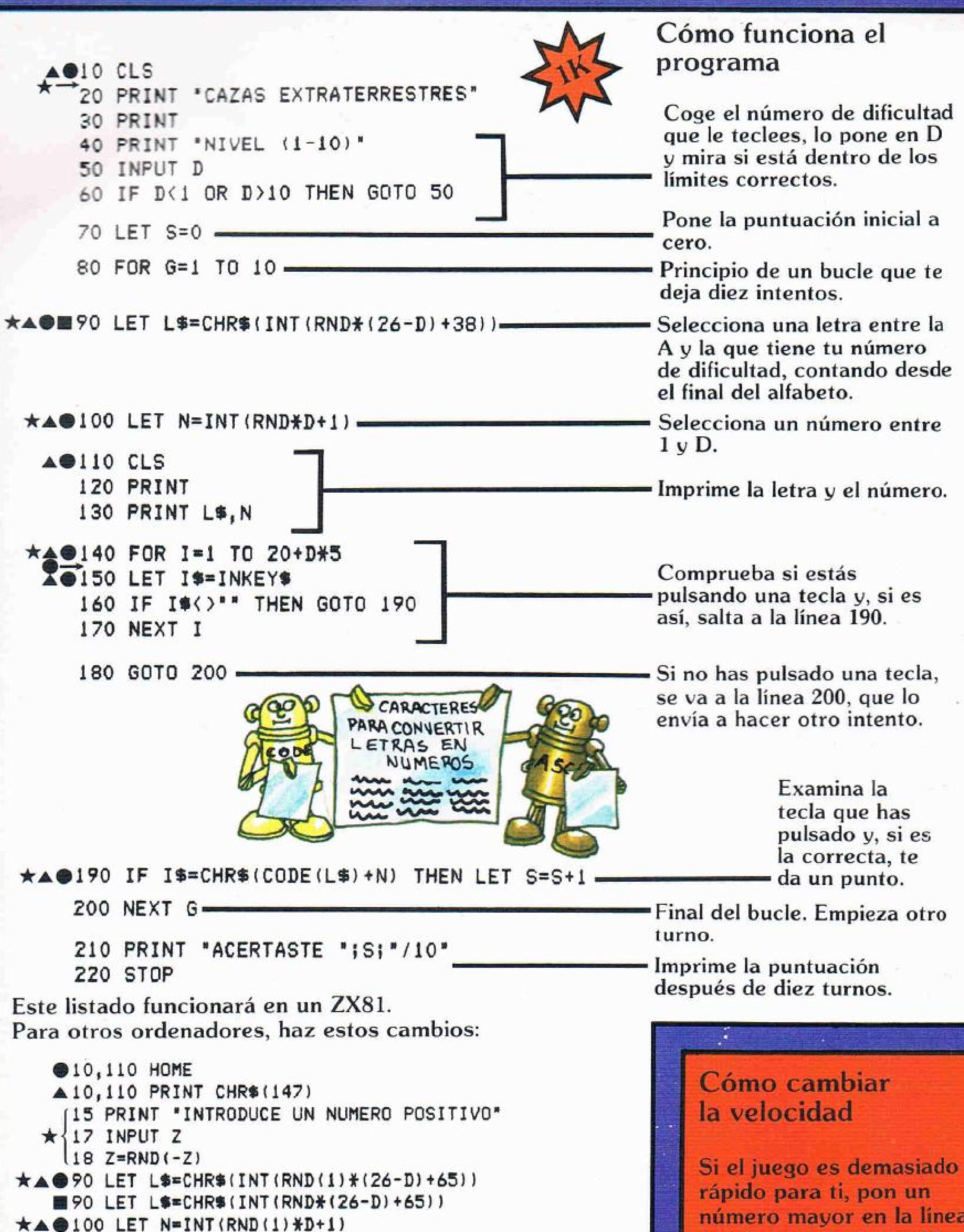


Eres el capitán de un crucero interestelar que, dañado en uno de sus motores, ha entrado en un área prohibida. Los cazas extraterrestres te están atacando y, para empeorar las cosas, están utilizando un extraño dispositivo contra tu radar que hace que dé falsas lecturas. Afortunadamente, tu ordenador conoce un código que puedes usar para hallar las posiciones correctas de las naves enemigas. Pero debes ser rápido. ¡No permanecen en el mismo sitio durante mucho tiempo!

Tu ordenador imprimirá una letra (la posición falsa del enemigo) y un número de código. Debes contar en el alfabeto ese número de letras a partir de la que te da (recuerda que la Ñ no existe) y pulsar la letra que obtengas. Por ejemplo, para M4 debes teclear una Q y para C2 una E, etc. Al pulsar la letra, dispararás automáticamente tu cañón láser, de modo que si era correcta obtendrás un acierto. Puedes elegir la dificultad en cada juego. Ésta viene dada por un número del 1 al 10 y es el máximo número de letras que se puede añadir cada vez. En cada juego hay diez cazas. ¡A ver cuántos puedes destruir!

▲★●■ *mira la página 2.*

Cómo funciona el programa




```
 ▲●10 CLS
★→20 PRINT "CAZAS EXTRATERRESTRES"
   30 PRINT
   40 PRINT "NIVEL (1-10)"
   50 INPUT D
   60 IF D<1 OR D>10 THEN GOTO 50
   70 LET S=0
   80 FOR G=1 TO 10
★▲●■90 LET L$=CHR$(INT(RND*(26-D)+38))
★▲●100 LET N=INT(RND*D+1)
 ▲●110 CLS
   120 PRINT
   130 PRINT L$,N
★▲●140 FOR I=1 TO 20+D*5
●→▲●150 LET I$=INKEY$
   160 IF I$<>"" THEN GOTO 190
   170 NEXT I
   180 GOTO 200
★▲●190 IF I$=CHR$(CODE(L$)+N) THEN LET S=S+1
   200 NEXT G
   210 PRINT "ACERTASTE ";S;"/10"
   220 STOP
```

10–60: Coge el número de dificultad que le teclees, lo pone en D y mira si está dentro de los límites correctos.

70: Pone la puntuación inicial a cero.

80: Principio de un bucle que te deja diez intentos.

90: Selecciona una letra entre la A y la que tiene tu número de dificultad, contando desde el final del alfabeto.

100: Selecciona un número entre 1 y D.

110–130: Imprime la letra y el número.

140–170: Comprueba si estás pulsando una tecla y, si es así, salta a la línea 190.

180: Si no has pulsado una tecla, se va a la línea 200, que lo envía a hacer otro intento.

190: Examina la tecla que has pulsado y, si es la correcta, te da un punto.

200: Final del bucle. Empieza otro turno.

210–220: Imprime la puntuación después de diez turnos.

Este listado funcionará en un ZX81.
Para otros ordenadores, haz estos cambios:

```
 ●10,110 HOME
 ▲10,110 PRINT CHR$(147)
  ⎧15 PRINT "INTRODUCE UN NUMERO POSITIVO"
 ★⎨17 INPUT Z
  ⎩18 Z=RND(-Z)
★▲●90 LET L$=CHR$(INT(RND(1)*(26-D)+65))
  ■90 LET L$=CHR$(INT(RND*(26-D)+65))
★▲●100 LET N=INT(RND(1)*D+1)
 ★▲140 FOR I=1 TO 200+D*50
  ●140 FOR I=1 TO 100+D*50
  ●145 I$=""
  ▲150 GET I$
  ●150 IF PEEK(-16384)>127 THEN GET I$
★▲●190 IF I$=CHR$(ASC(L$)+N) THEN LET S=S+1
```

Cómo cambiar la velocidad

Si el juego es demasiado rápido para ti, pon un número mayor en la línea 140 (p. ej., sustituye 20 por 200). Puedes hacerlo más rápido con un número menor.

Cómo hacer el juego más difícil

Puedes cambiar el 1 en las líneas 40 y 50, a 3 por ejemplo, para permitir dificultades de 3 o más.

Rincón de los problemas

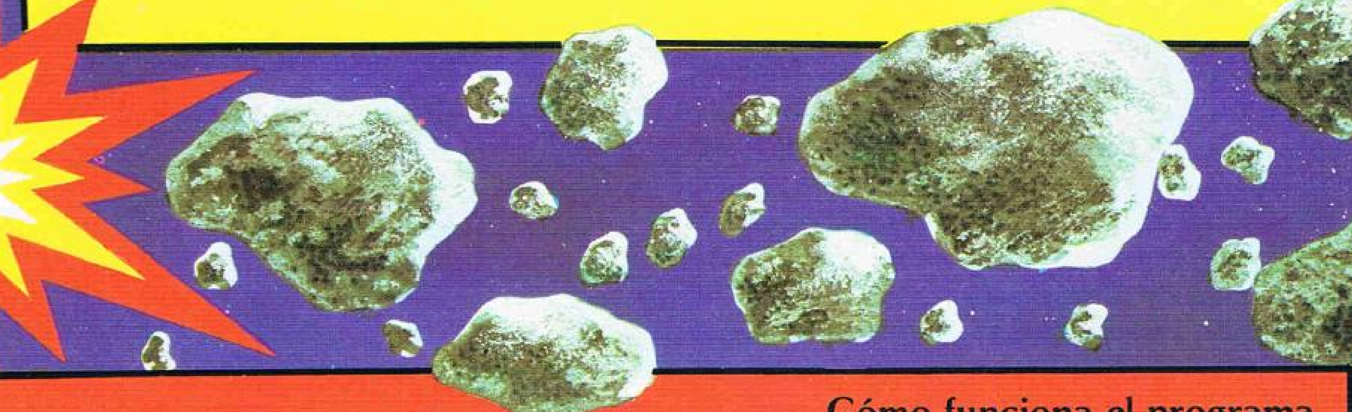
¿Puedes ajustar la puntuación de modo que te dé los mismos puntos que indique el código? Es decir, el código 1 da un punto, el código 2 da 2 puntos, etc.



Cinturón de asteroides

Estás realizando un viaje a través del cinturón de asteroides, y para no chocar con ellos, debes destruirlos. La fuerza necesaria para ello depende del tamaño de cada uno.

Los asteroides aparecen en tu pantalla como grupos de estrellas. Para destruirlos, debes pulsar la tecla del número correspondiente al número de estrellas (por ejemplo, si hay siete estrellas, debes pulsar un 7). ¡Prepárate: los asteroides vienen hacia ti, duros y rápidos!



Cómo funciona el programa

```
    10 PRINT "CINTURON DE ASTEROIDES"
★→  20 LET S=0
    30 FOR G=1 TO 10
▲●  40 CLS
★▲● 50 LET A=INT(RND*18+1)
★▲● 60 LET D=INT(RND*12+1)
★▲● 70 LET N=INT(RND*9+1)
    80 FOR I=1 TO D
    90 PRINT
    100 NEXT I
    110 FOR I=1 TO N
    120 IF I<>1 AND I<>4 AND I<>7 THEN GOTO 150
    130 PRINT
    140 PRINT TAB(A);
    150 PRINT "*";
    160 NEXT I
●→  170 PRINT
★▲● 180 FOR I=1 TO 10
▲●  190 LET Q=VAL("0"+INKEY$)
    200 IF Q<>0 THEN GOTO 240
    210 NEXT I
    220 PRINT "CHOCASTE CON UN ASTEROIDE"
    230 GOTO 290
    240 IF Q<>N THEN GOTO 270
```

- 20: Pone la puntuación inicial a cero.
- 30: Empieza un bucle que se repite diez veces.
- 50: Elige un número para la posición horizontal del asteroide en la pantalla. Almacena dicho número en A.
- 60: Elige otro número (del 1 al 12) para la posición vertical, y lo pone en D.
- 70: Se elige un tercer número (del 1 al 9) para el número de estrellas del asteroide.
- 80–100: Mueve el cursor D líneas hacia abajo.
- 110–160: Ejecuta este bucle N veces imprimiendo una estrella cada vez en la posición apropiada.
- 180–210: Hace un bucle para ver si pulsas una tecla y, si es así, salta a la línea 240.
- 220–230: Imprime esto si te quedas sin tiempo.
- 240: Comprueba si el número que has pulsado es el mismo que N y, si no, salta a la línea 270.

Programa	Explicación
`250 PRINT "LO DESTRUISTE"`	Imprime esto si pulsaste el número adecuado.
`260 LET S=S+1`	Incrementa tu puntuación en 1.
`270 IF Q<N THEN PRINT "NO SUFICIENTEMENTE FUERTE"` `280 IF Q>N THEN PRINT "DEMASIADO FUERTE"`	Compara tu número con N e imprime el mensaje apropiado.
`★▲●290 FOR I=1 TO 50` `300 NEXT I`	Bucle de espera para que los mensajes permanezcan en pantalla el tiempo suficiente para ser leídos.
`310 NEXT G`	Vuelve atrás por otro intento.
`320 PRINT "ACERTASTE ";S;" DE 10"` `330 STOP`	Imprime la puntuación después de 10 intentos.

Este listado funcionará en un ZX81.
Para otros ordenadores, haz estos cambios:

```
  ★ {15 PRINT "INTRODUCE UN NUMERO POSITIVO"
    {17 INPUT Z
    {18 Z=RND(-Z)
    ●40 HOME
    ▲40 PRINT CHR$(147)
  ★▲●50 LET A=INT(RND(1)*18+1)
  ★▲●60 LET D=INT(RND(1)*12+1)
  ★▲●70 LET N=INT(RND(1)*9+1)
    ●175 Q=0
  ★▲●180 FOR I=1 TO 100
    ▲190 GET Q
    ●190 IF PEEK(-16384)>127 THEN GET Q
  ★▲●290 FOR I=1 TO 250
```

Como cambiar la velocidad

La línea 180 controla cuánto tiempo tienes para pulsar una tecla. Cambia el último número en esa línea por uno inferior para acelerar el juego.

Rincón de los problemas

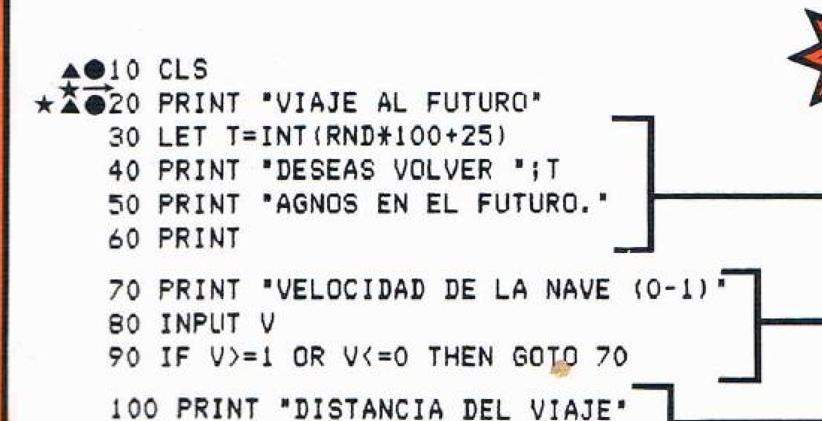


Adapta la puntuación de modo que para cada asteroide obtengas el mismo número de puntos que estrellas hay en él?

▲★●■ *mira la página 2.*

Viaje al futuro

Vas en una nave espacial que viaja casi tan veloz como la luz. Curiosamente, el tiempo pasa más despacio dentro de tu nave que fuera, de modo que si emprendes un largo y rápido viaje, puedes volver a la Tierra mucho más allá del futuro que lo que indique el reloj de tu nave. En este juego, tu ordenador te dice cuántos años deben pasar en la Tierra antes de que vuelvas. Tienes que decidir la longitud de tu viaje (en años-luz) y la velocidad de tu nave (como una fracción de la de la luz). ¡Si vas demasiado lejos o demasiado despacio, morirás de viejo en el camino!



```
▲●10 CLS
★▲●20 PRINT "VIAJE AL FUTURO"
   30 LET T=INT(RND*100+25)
   40 PRINT "DESEAS VOLVER ";T
   50 PRINT "AGNOS EN EL FUTURO."
   60 PRINT
   70 PRINT "VELOCIDAD DE LA NAVE (0-1)"
   80 INPUT V
   90 IF V>=1 OR V<=0 THEN GOTO 70
  100 PRINT "DISTANCIA DEL VIAJE"
  110 INPUT D
  120 LET T1=D/V
```

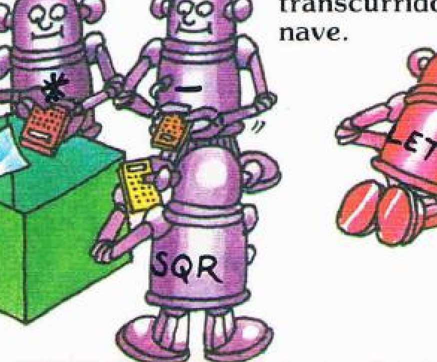
Cómo funciona el programa

Líneas 30–60: Elige un número entre 25 y 124, que es el que corresponde a los años que deben pasar antes de que vuelvas, y lo imprime.

Líneas 70–90: Te pide la velocidad y comprueba que está dentro de los límites correctos.

Líneas 100–110: Te pide la distancia.

Línea 120: Calcula el tiempo transcurrido dentro de la nave.



▲★●■ mira la página 2.

```
130 LET T2=T1/SQR(1-V*V)
```

Calcula el tiempo transcurrido fuera de la nave (es decir, en la Tierra).

```
140 PRINT "TARDASTE ";T1;" AGNOS"
150 PRINT "Y LLEGASTE ";T2;" AGNOS"
160 PRINT "EN EL FUTURO."
```

Imprime estos tiempos.

```
170 IF T1>50 THEN GOTO 210
```

Comprueba si tardaste más que tu tiempo de vida (50 años) y salta a la línea 210, si es así.

```
180 IF ABS(T-T2)<=5 THEN PRINT "LLEGASTE A TIEMPO"
190 IF ABS(T-T2)>5 THEN PRINT "NI SIQUIERA CERCA"
200 STOP
210 PRINT "HAS MUERTO EN EL CAMINO"
220 STOP
```

Comprueba si acertaste con un error de 5 años e imprime un mensaje.

Este listado funcionará en un ZX81.
Para otros ordenadores, haz estos cambios:

```
 ●10 HOME
 ▲10 PRINT CHR$(147)
  {15 PRINT "INTRODUCE UN NUMERO POSITIVO"
★{17 INPUT Z
  {18 Z=RND(-Z)
★▲●30 LET T=INT(RND(1)*100+25)
```

Rincón de los problemas



¿Puedes cambiar el programa para que haga las siguientes cosas?:

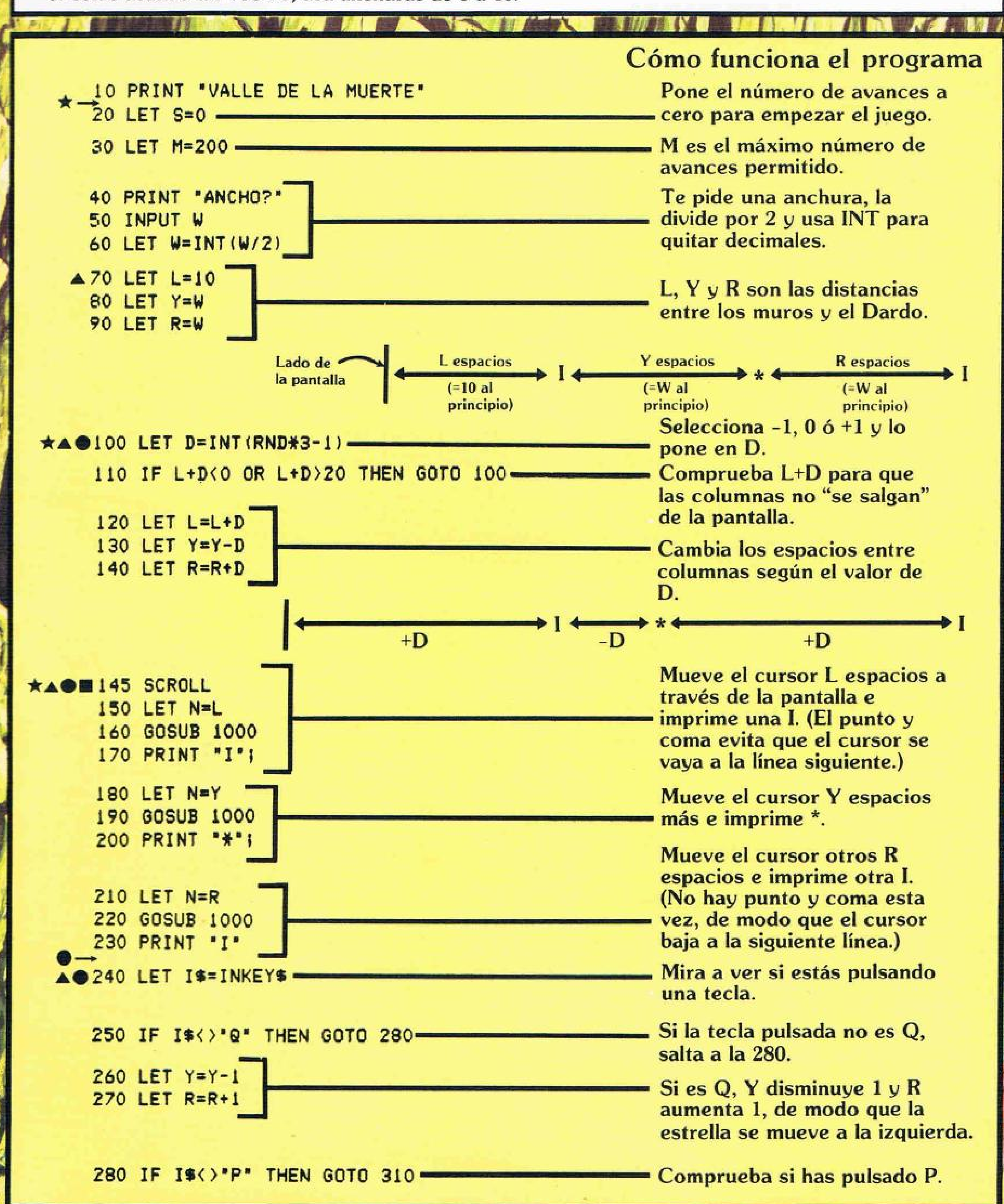
1) Dar un rango más amplio de años que deben pasar antes de volver a la Tierra.
2) Disminuir el error permitido de 5 años a 2 años.
3) Modificar tu tiempo de vida.

El valle de la Muerte

Sólo hay un método de escapar a las fuerzas de los malvados Disectitrones. Tendrás que tensar todos tus nervios y guiar tu nave monoplaza Dardo por el estrecho desfiladero sin fondo conocido como valle de la Muerte.

Tu ordenador te pedirá primero la anchura del valle. Intenta primero 15* y, después, ve disminuyendo; ocho ya es bastante difícil. Mueve tu nave pulsando Q para la izquierda y P para la derecha, e intenta atravesar sano y salvo el valle de la Muerte.

* Si estás usando un VIC 20, usa anchuras de 6 a 10.



Cómo funciona el programa

```
   10 PRINT "VALLE DE LA MUERTE"
★→ 20 LET S=0
```
Pone el número de avances a cero para empezar el juego.

```
30 LET M=200
```
M es el máximo número de avances permitido.

```
40 PRINT "ANCHO?"
50 INPUT W
60 LET W=INT(W/2)
```
Te pide una anchura, la divide por 2 y usa INT para quitar decimales.

```
▲70 LET L=10
 80 LET Y=W
 90 LET R=W
```
L, Y y R son las distancias entre los muros y el Dardo.


```
★▲●100 LET D=INT(RND*3-1)
```
Selecciona -1, 0 ó +1 y lo pone en D.

```
110 IF L+D<0 OR L+D>20 THEN GOTO 100
```
Comprueba L+D para que las columnas no "se salgan" de la pantalla.

```
120 LET L=L+D
130 LET Y=Y-D
140 LET R=R+D
```
Cambia los espacios entre columnas según el valor de D.


```
★▲●■145 SCROLL
    150 LET N=L
    160 GOSUB 1000
    170 PRINT "I";
```
Mueve el cursor L espacios a través de la pantalla e imprime una I. (El punto y coma evita que el cursor se vaya a la línea siguiente.)

```
180 LET N=Y
190 GOSUB 1000
200 PRINT "*";
```
Mueve el cursor Y espacios más e imprime *.

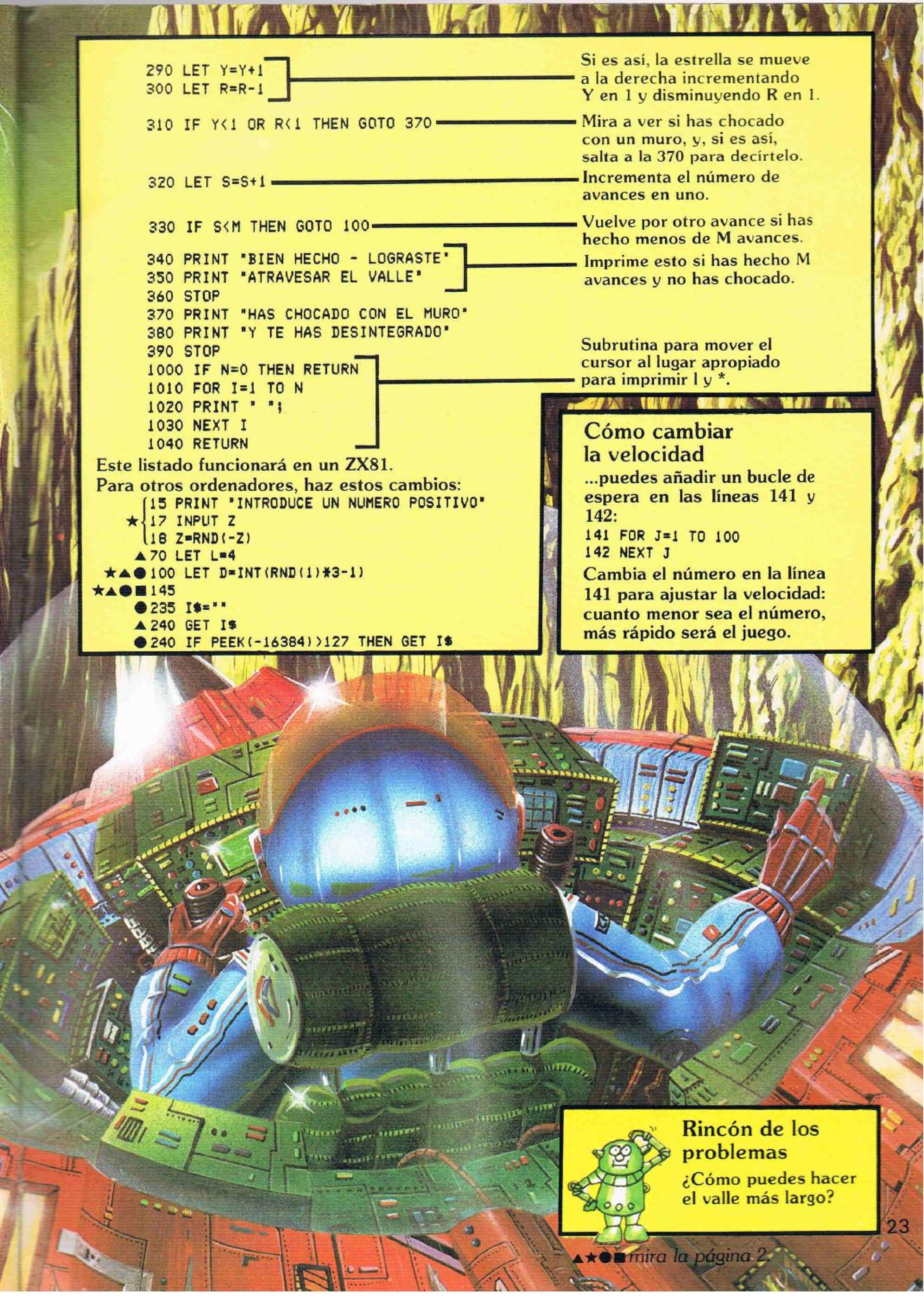
```
210 LET N=R
220 GOSUB 1000
230 PRINT "I"
```
Mueve el cursor otros R espacios e imprime otra I. (No hay punto y coma esta vez, de modo que el cursor baja a la siguiente línea.)

```
●→
▲●240 LET I$=INKEY$
```
Mira a ver si estás pulsando una tecla.

```
250 IF I$<>"Q" THEN GOTO 280
```
Si la tecla pulsada no es Q, salta a la 280.

```
260 LET Y=Y-1
270 LET R=R+1
```
Si es Q, Y disminuye 1 y R aumenta 1, de modo que la estrella se mueve a la izquierda.

```
280 IF I$<>"P" THEN GOTO 310
```
Comprueba si has pulsado P.



```
290 LET Y=Y+1
300 LET R=R-1
```

Si es así, la estrella se mueve a la derecha incrementando Y en 1 y disminuyendo R en 1.

```
310 IF Y<1 OR R<1 THEN GOTO 370
```

Mira a ver si has chocado con un muro, y, si es así, salta a la 370 para decírtelo.

```
320 LET S=S+1
```

Incrementa el número de avances en uno.

```
330 IF S<M THEN GOTO 100
```

Vuelve por otro avance si has hecho menos de M avances.

```
340 PRINT "BIEN HECHO - LOGRASTE"
350 PRINT "ATRAVESAR EL VALLE"
```

Imprime esto si has hecho M avances y no has chocado.

```
360 STOP
370 PRINT "HAS CHOCADO CON EL MURO"
380 PRINT "Y TE HAS DESINTEGRADO"
390 STOP
1000 IF N=0 THEN RETURN
1010 FOR I=1 TO N
1020 PRINT " ";
1030 NEXT I
1040 RETURN
```

Subrutina para mover el cursor al lugar apropiado para imprimir I y *.

Este listado funcionará en un ZX81.
Para otros ordenadores, haz estos cambios:

```
    {15 PRINT "INTRODUCE UN NUMERO POSITIVO"
   ★{17 INPUT Z
    {18 Z=RND(-Z)
    ▲70 LET L=4
  ★▲●100 LET D=INT(RND(1)*3-1)
 ★▲●■145
    ●235 I$=""
    ▲240 GET I$
    ●240 IF PEEK(-16384)>127 THEN GET I$
```

Cómo cambiar la velocidad

...puedes añadir un bucle de espera en las líneas 141 y 142:

```
141 FOR J=1 TO 100
142 NEXT J
```

Cambia el número en la línea 141 para ajustar la velocidad: cuanto menor sea el número, más rápido será el juego.

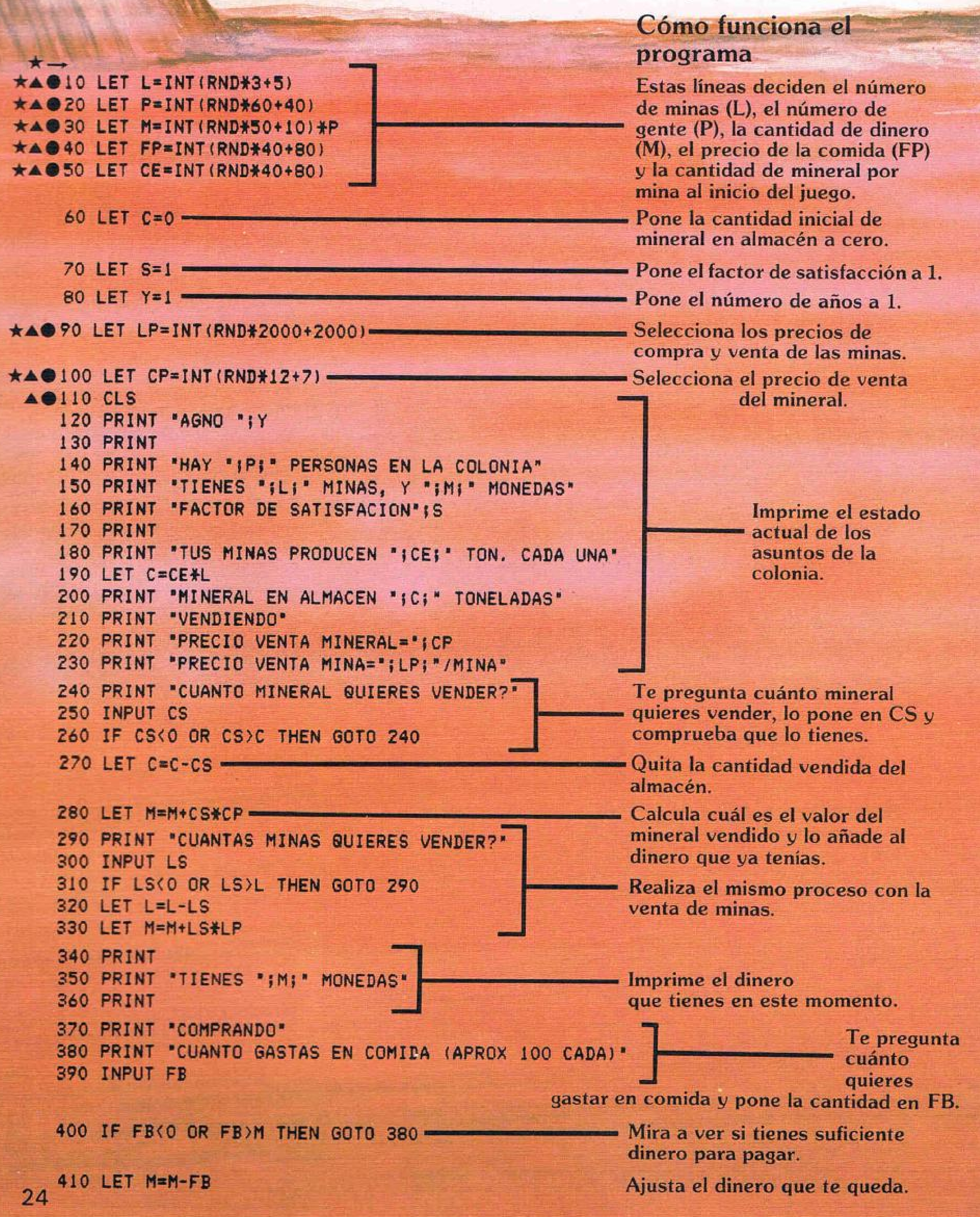
Rincón de los problemas

¿Cómo puedes hacer el valle más largo?

▲★●■ *mira la página 2.*

Minas espaciales

Eres el nuevo líder electo de una colonia minera en el planeta Astron. Todas las decisiones relativas a la venta de mineral a los comerciantes intergalácticos, compra de comida y venta y compra de minas son hechas por ti. ¿Puedes mantener a la gente satisfecha y sobrevivir 10 años al mando? ¿O acabará la vida de la colonia en un desastre bajo tu mandato?



Cómo funciona el programa

```
★→
★▲●10 LET L=INT(RND*3+5)
★▲●20 LET P=INT(RND*60+40)
★▲●30 LET M=INT(RND*50+10)*P
★▲●40 LET FP=INT(RND*40+80)
★▲●50 LET CE=INT(RND*40+80)
```

Estas líneas deciden el número de minas (L), el número de gente (P), la cantidad de dinero (M), el precio de la comida (FP) y la cantidad de mineral por mina al inicio del juego.

```
60 LET C=0
```

Pone la cantidad inicial de mineral en almacén a cero.

```
70 LET S=1
```

Pone el factor de satisfacción a 1.

```
80 LET Y=1
```

Pone el número de años a 1.

```
★▲●90 LET LP=INT(RND*2000+2000)
```

Selecciona los precios de compra y venta de las minas.

```
★▲●100 LET CP=INT(RND*12+7)
```

Selecciona el precio de venta del mineral.

```
 ▲●110 CLS
120 PRINT "AGNO ";Y
130 PRINT
140 PRINT "HAY ";P;" PERSONAS EN LA COLONIA"
150 PRINT "TIENES ";L;" MINAS, Y ";M;" MONEDAS"
160 PRINT "FACTOR DE SATISFACION";S
170 PRINT
180 PRINT "TUS MINAS PRODUCEN ";CE;" TON. CADA UNA"
190 LET C=CE*L
200 PRINT "MINERAL EN ALMACEN ";C;" TONELADAS"
210 PRINT "VENDIENDO"
220 PRINT "PRECIO VENTA MINERAL=";CP
230 PRINT "PRECIO VENTA MINA=";LP;"/MINA"
```

Imprime el estado actual de los asuntos de la colonia.

```
240 PRINT "CUANTO MINERAL QUIERES VENDER?"
250 INPUT CS
260 IF CS<0 OR CS>C THEN GOTO 240
```

Te pregunta cuánto mineral quieres vender, lo pone en CS y comprueba que lo tienes.

```
270 LET C=C-CS
```

Quita la cantidad vendida del almacén.

```
280 LET M=M+CS*CP
```

Calcula cuál es el valor del mineral vendido y lo añade al dinero que ya tenías.

```
290 PRINT "CUANTAS MINAS QUIERES VENDER?"
300 INPUT LS
310 IF LS<0 OR LS>L THEN GOTO 290
320 LET L=L-LS
330 LET M=M+LS*LP
```

Realiza el mismo proceso con la venta de minas.

```
340 PRINT
350 PRINT "TIENES ";M;" MONEDAS"
360 PRINT
```

Imprime el dinero que tienes en este momento.

```
370 PRINT "COMPRANDO"
380 PRINT "CUANTO GASTAS EN COMIDA (APROX 100 CADA)"
390 INPUT FB
```

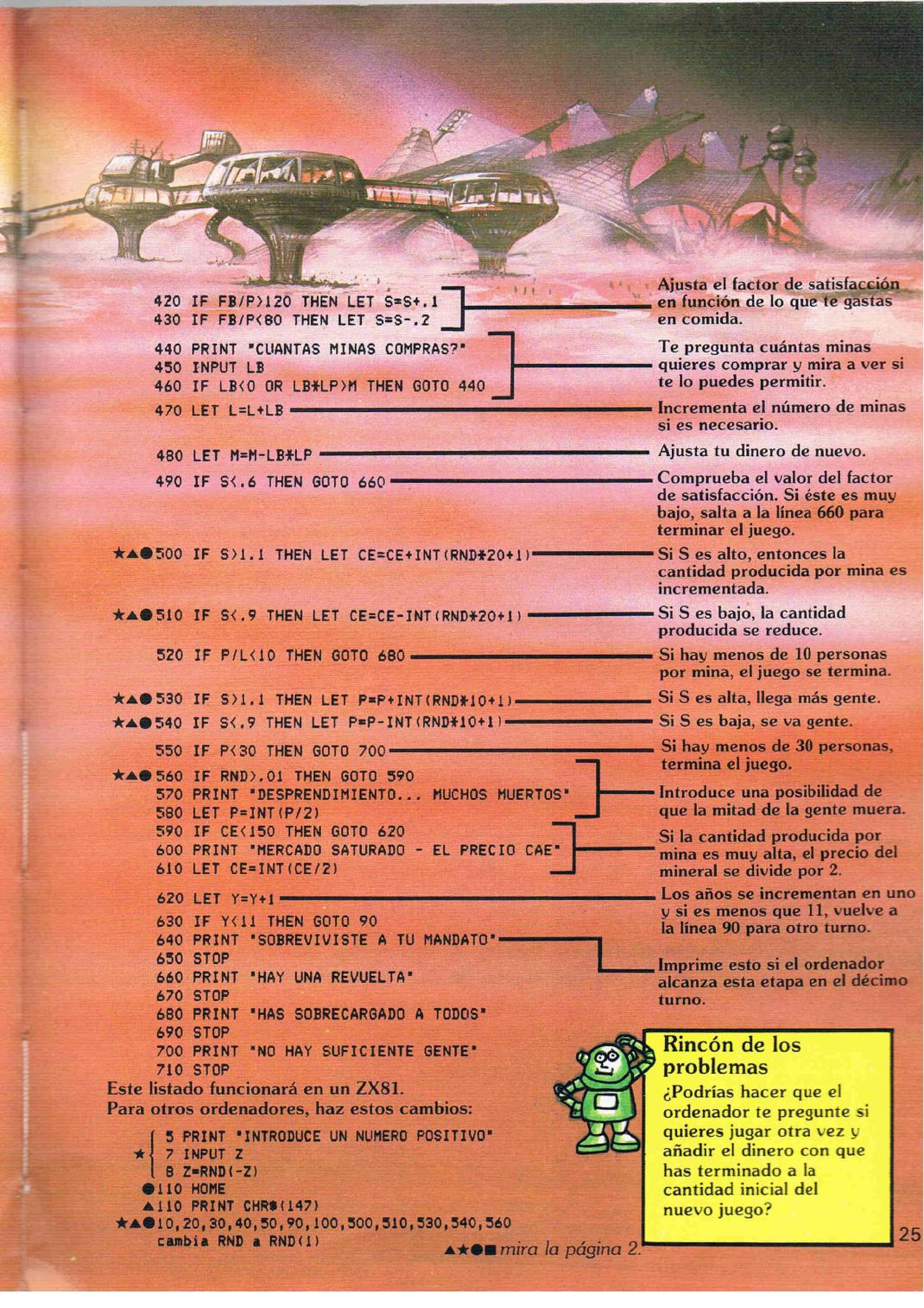
Te pregunta cuánto quieres gastar en comida y pone la cantidad en FB.

```
400 IF FB<0 OR FB>M THEN GOTO 380
```

Mira a ver si tienes suficiente dinero para pagar.

```
410 LET M=M-FB
```

Ajusta el dinero que te queda.



```
420 IF FB/P>120 THEN LET S=S+.1
430 IF FB/P<80 THEN LET S=S-.2
440 PRINT "CUANTAS MINAS COMPRAS?"
450 INPUT LB
460 IF LB<0 OR LB*LP>M THEN GOTO 440
470 LET L=L+LB
480 LET M=M-LB*LP
490 IF S<.6 THEN GOTO 660
★▲●500 IF S>1.1 THEN LET CE=CE+INT(RND*20+1)
★▲●510 IF S<.9 THEN LET CE=CE-INT(RND*20+1)
520 IF P/L<10 THEN GOTO 680
★▲●530 IF S>1.1 THEN LET P=P+INT(RND*10+1)
★▲●540 IF S<.9 THEN LET P=P-INT(RND*10+1)
550 IF P<30 THEN GOTO 700
★▲●560 IF RND>.01 THEN GOTO 590
570 PRINT "DESPRENDIMIENTO... MUCHOS MUERTOS"
580 LET P=INT(P/2)
590 IF CE<150 THEN GOTO 620
600 PRINT "MERCADO SATURADO - EL PRECIO CAE"
610 LET CE=INT(CE/2)
620 LET Y=Y+1
630 IF Y<11 THEN GOTO 90
640 PRINT "SOBREVIVISTE A TU MANDATO"
650 STOP
660 PRINT "HAY UNA REVUELTA"
670 STOP
680 PRINT "HAS SOBRECARGADO A TODOS"
690 STOP
700 PRINT "NO HAY SUFICIENTE GENTE"
710 STOP
```

- 420–430: Ajusta el factor de satisfacción en función de lo que te gastas en comida.
- 440–460: Te pregunta cuántas minas quieres comprar y mira a ver si te lo puedes permitir.
- 470: Incrementa el número de minas si es necesario.
- 480: Ajusta tu dinero de nuevo.
- 490: Comprueba el valor del factor de satisfacción. Si éste es muy bajo, salta a la línea 660 para terminar el juego.
- 500: Si S es alto, entonces la cantidad producida por mina es incrementada.
- 510: Si S es bajo, la cantidad producida se reduce.
- 520: Si hay menos de 10 personas por mina, el juego se termina.
- 530: Si S es alta, llega más gente.
- 540: Si S es baja, se va gente.
- 550: Si hay menos de 30 personas, termina el juego.
- 560–580: Introduce una posibilidad de que la mitad de la gente muera.
- 590–610: Si la cantidad producida por mina es muy alta, el precio del mineral se divide por 2.
- 620–630: Los años se incrementan en uno y si es menos que 11, vuelve a la línea 90 para otro turno.
- 640: Imprime esto si el ordenador alcanza esta etapa en el décimo turno.

Este listado funcionará en un ZX81.
Para otros ordenadores, haz estos cambios:

```
  { 5 PRINT "INTRODUCE UN NUMERO POSITIVO"
★ { 7 INPUT Z
  { 8 Z=RND(-Z)
●110 HOME
▲110 PRINT CHR$(147)
★▲●10,20,30,40,50,90,100,500,510,530,540,560
    cambia RND a RND(1)
```

▲★●■ *mira la página 2.*

Rincón de los problemas

¿Podrías hacer que el ordenador te pregunte si quieres jugar otra vez y añadir el dinero con que has terminado a la cantidad inicial del nuevo juego?

Rescate espacial

Debes hacer un viaje urgente, a través del brazo espiral de la galaxia, a un planeta en desarrollo que necesita suministros médicos. El viaje supone recorrer distancias tan largas que la mayor parte del tiempo estarás en un sueño profundo; pero, antes de esto, debes programar la nave para el viaje. El ordenador te preguntará cuánta energía quieres proporcionar a los motores, al soporte de vida y a los escudos antes de empezar a dormir.

Cuando te despiertes te dará un informe de lo que ha sucedido durante el viaje y, si todo va bien, estarás orbitando alrededor del planeta. En este momento, debes utilizar tu energía restante en los cohetes de aterrizaje y en los escudos, de modo que puedas hacer un buen aterrizaje en el planeta.

Si logras terminar la misión a salvo, tienes una buena posibilidad de que te asciendan a almirante espacial. ¡Buena suerte!

```
 ▲●10 CLS
 ★→20 PRINT "RESCATE ESPACIAL"
   30 PRINT
   40 PRINT "QUIERES INSTRUCCIONES?"
   50 INPUT I$
★▲●60 IF I$(1)="S" THEN GOSUB 1000
★▲●70 LET D=INT(RND*800+101)
★▲●80 LET E=INT(RND*400+401)
   90 LET T=INT(D/SQR(E/5)+.5)
  100 PRINT "EL PLANETA ESTA A ";D;" UNIDADES"
  110 PRINT "TIENES ";E;" UNIDADES DE ENERGIA"
  120 PRINT "Y UN TIEMPO LIMITE DE ";T;" DIAS"
  130 PRINT
  140 PRINT "DISTRIBUCION DE ENERGIA:"
  150 PRINT "MOTORES?"
  160 INPUT P
  170 PRINT "SOPORTE DE VIDA?"
  180 INPUT L
  190 PRINT "ESCUDOS?"
  200 INPUT S
  210 IF P+L+S>E THEN GOTO 140
  220 LET X=E-P-L-S
  230 LET V=INT(SQR(P))
  240 LET T1=INT(D/V)
 ▲●250 CLS
  260 PRINT "TU VELOCIDAD ES ";V
  270 PRINT "LLEGADA EN ";T1;" DIAS"
  280 PRINT
★▲●290 FOR I=1 TO INT(RND*5+6)
★▲●300 IF RND>.5 THEN GOTO 430
★▲●310 GOTO 320+INT(RND*4)*30
  320 PRINT "TORMENTA DE ASTEROIDES - ESCUDOS ROTOS"
```

```
★▲●330 LET S=S-20-INT(RND*40+1)
   340 GOTO 430
   350 PRINT "ORDENADOR ROTO:RETARDO EN REPARACION"
★▲●360 LET D=D+INT(RND*20+1)
   370 GOTO 430
   380 PRINT "PROBLEMA DE MOTOR: DEBEMOS FRENAR"
   390 LET V=V-.5
   400 GOTO 430
   410 PRINT "RAYOS X - SOPORTE DE VIDA ROTO"
   420 LET L=L-20-INT(RND*40+1)
★▲●430 FOR J=1 TO 50
   440 NEXT J
   450 NEXT I
   460 LET T1=INT(D/V)
 ▲●470 CLS
   480 PRINT "LLEGADA EN ";T1;" DIAS"
   490 IF S<0 THEN PRINT "ESCUDOS ROTOS"
   495 IF S<0 THEN PRINT "FUISTE DESTRUIDO"
   500 IF L<=0 THEN PRINT "SOPORTE DE VIDA INACTIVO"
   505 IF L<=0 THEN PRINT "ESTAS MUERTO"
   510 IF V<=0 THEN PRINT "MOTORES NO USABLES"
   520 IF T1>T THEN PRINT "TARDASTE DEMASIADO"
   530 IF S<0 OR L<=0 OR V<=0 OR T1>T THEN STOP
★▲●540 LET G=INT(RND*10+5)
   550 LET G$="ALTA"
   560 IF G<12 THEN LET G$= "MEDIA"
   570 IF G<8 THEN LET G$="BAJA"
★▲●580 LET A=INT(RND*10+5)
   590 LET A$="ALTA"
   600 IF A<12 THEN LET A$="MEDIA"
   610 IF A<8 THEN LET A$="BAJA"
   620 PRINT
   630 PRINT "ESTAS EN ORBITA DEL PLANETA"
   640 PRINT "ENERGIA DE SOBRA=";X
   650 PRINT "GRAVEDAD ES ";G$
   660 PRINT "ATMOSFERA ES ";A$
   670 PRINT
   680 PRINT "CUANTA ENERGIA A COHETES-FRENO?"
   690 INPUT B
   700 PRINT "CUANTA ENERGIA A ESCUDOS?"
   710 INPUT S
   720 IF B+S>X THEN GOTO 680
 ▲●730 CLS
   740 IF B>=G*10 THEN GOTO 770
```

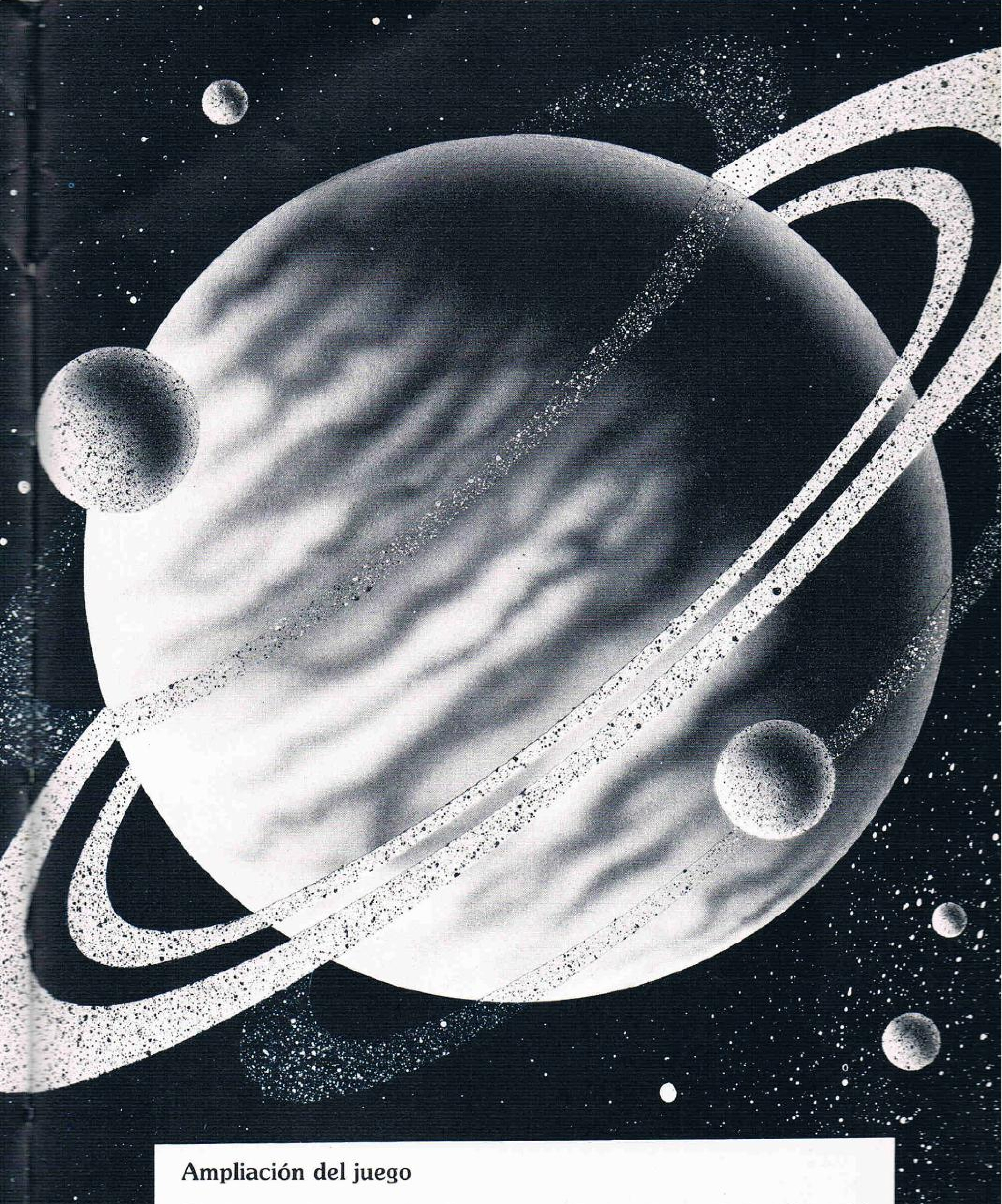
El programa continúa en la página siguiente.

Continuación de Rescate espacial

```
750 PRINT "HAS HECHO UN NUEVO CRATER"
760 GOTO 840
770 IF S>=A*10 THEN GOTO 800
780 PRINT "HAS CREADO UNA ESTRELLA FUGAZ"
790 GOTO 840
800 PRINT "ATERRIZASTE BIEN - ENHORABUENA"
810 IF X-S-B>25 THEN GOTO 840
820 PRINT "ES UNA PENA QUE NO TENGAS"
830 PRINT "ENERGIA PARA ABRIR LA PUERTA"
840 STOP
1000 PRINT
1010 PRINT "ESTAS A PUNTO DE EMBARCAR"
1020 PRINT "PARA UNA MISION A UN "
1030 PRINT "DISTANTE PLANETA QUE NECESITA"
1040 PRINT "URGENTEMENTE MEDICINAS."
1050 PRINT "PRIMERO DEBES PREPARAR LA NACE"
1060 PRINT "PARA EL VIAJE DISTRIBUYENDO"
1070 PRINT "LA ENERGIA A LOS MOTORES,"
1080 PRINT "ESCUDOS Y SOPORTE DE VIDA."
1090 PRINT "ENTONCES DORMIRAS DURANTE EL"
1100 PRINT "VIAJE Y AL DESPERTAR RECIBIRAS"
1110 PRINT "UN INFORME DE LO SUCEDIDO."
1120 PRINT "A CONTINUACION DEBES ATERRIZAR"
1130 PRINT "EN EL PLANETA..."
1140 PRINT "PULSA UNA TECLA"
▲●1150 IF INKEY$="" THEN GOTO 1150
▲●1160 CLS
1170 RETURN
```

Este listado funcionará en un ZX81.
Para otros ordenadores, haz estos cambios:

```
★▲●cambiar todos los RND a RND(1)
  ⎡15 PRINT "INTRODUCE UN NUMERO POSITIVO"
★ ⎨17 INPUT Z
  ⎣18 Z=RND(-Z)
●10,250,470,730,1160 HOME
▲10,250,470,730,1160 PRINT CHR$(147)
★▲●60 IF LEFT$(I$,1)="S" THEN GOSUB 1000
★▲●310 ON INT(RND*4+1) GOTO 320,350,380,410
★▲●430 FOR J=1 TO 500
●1150 GET I$
▲1150 GET I$:IF I$="" THEN GOTO 1150
```



Ampliación del juego

Este juego realmente consta de dos partes. En la primera, fijas los datos del viaje con el objetivo de ponerte en órbita alrededor del planeta, y en la segunda, intentas aterrizar en el planeta. Podrías intentar añadir una tercera parte en la que haga el peligroso recorrido desde el lugar de aterrizaje hasta el cuartel general de la Cruz Roja Intergaláctica.

Contacto

Este juego es diferente de los otros del libro, debido a que usa gráficos. Como los ordenadores varían tanto en el modo en que sus gráficos trabajan, hay un programa separado para cada uno. Lee las instrucciones de esta página para aprender a jugar y, entonces, busca en las páginas siguientes la versión de tu ordenador.

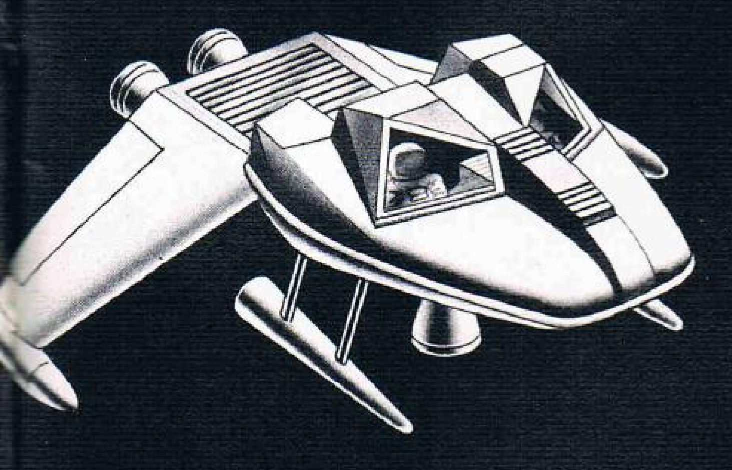
Cómo jugar

El as de los pilotos espaciales, el capitán Flash, está sentado a tu lado para realizar la parte final de tu examen de manejo avanzado de naves espaciales (parte III). Tu ligero módulo biplaza de aterrizaje se aproxima rápidamente a la superficie de la Luna, pero tu velocidad debe ser casi cero cuando toques tierra. Controla la fuerza con la tecla A para incrementarla y D para reducirla*, vigilando tu progreso en la pantalla todo el tiempo. Si usas demasiada fuerza empezarás a subir de nuevo; demasiado poca y harás un nuevo cráter en la Luna. ¿Podrás impresionar al capitán Flash con tu destreza?

* Para el VIC y el Commodore 64, usa la flecha hacia abajo para incrementar la fuerza y la flecha hacia la derecha para reducirla.

Contacto: versión para Commodore 64

```
20 PRINT CHR$(147)CHR$(5);
25 POKE 53281,0
30 DEF FNR(X)=INT(RND(1)*X+1)
40 GOSUB 250
50 GOSUB 300
60 C=1:GOSUB 390
70 A=1:B=F:GOSUB 460
80 A=2:B=ABS(V):GOSUB 460
90 A=3:B=H:GOSUB 460
100 A=4:B=T:GOSUB 460
110 GOSUB 530
120 V1=V-T/20+G:F=F-T/10
130 H1=H-(V+V1/10)
140 C=0:GOSUB 390
150 IF H1<10 THEN 200
160 H=H1:V=V1
170 IF H<=120 THEN 60
180 GOSUB 590
190 GOSUB 220
200 H=0:C=1:GOSUB 390
210 GOSUB 650
220 END
250 H=120:F=100:T=0
260 V=5+FNR(10)
270 G=(FNR(40)+40)/100
280 RETURN
300 FOR X=1998 TO 2047
320 POKE X,98+2*FNR(3)
330 NEXT
340 PRINT"GRAVEDAD=";G
350 PRINT "■■FUEL:"
355 PRINT"■■■VEL:"
360 PRINT"■■■ALTURA:"
365 PRINT "■■■FUERZA:"
370 RETURN
390 Y=1963-40*INT(H/5)
400 IF C=0 THEN 425
405 POKE Y,108:POKE Y+1,123
410 POKE Y+40,160:POKE Y+41,160
415 POKE Y+80,75:POKE Y+81,74
420 GOTO 440
425 FOR Z=0 TO 80 STEP 40
430 POKE Y+Z,32:POKE Y+Z+1,32
435 NEXT
440 RETURN
460 FOR X=0 TO 9
470 Y=A*160+X+1024
480 IF X<B/10 THEN POKE Y,102:GOTO 500
485 IF X-B/10+.5 THEN POKE Y,92:GOTO 500
490 POKE Y,32
500 NEXT
510 RETURN
530 GET I$
540 IF I$="■" THEN T=T+10:IF T>100
    THEN T=100
550 IF I$="■" THEN T=T-10:IF
    T<0 THEN T=0
560 IF T>F THEN T=F
570 RETURN
590 PRINT CHR$(147)
600 FOR I=1 TO 20
610 POKE 1063+FNR(506),42
620 NEXT
630 PRINT "PERDIDO EN EL ESPACIO!!"
640 RETURN
650 PRINT CHR$(147)"ATERRIZAJE"
660 PRINT "A LA VELOCIDAD:";
    INT((V+V1)*5)/10
670 IF (V+V1)<8 THEN PRINT
    "TODOS A SALVO":RETURN
680 PRINT "TODOS MUERTOS":RETURN
```



Contacto: versión para VIC 20

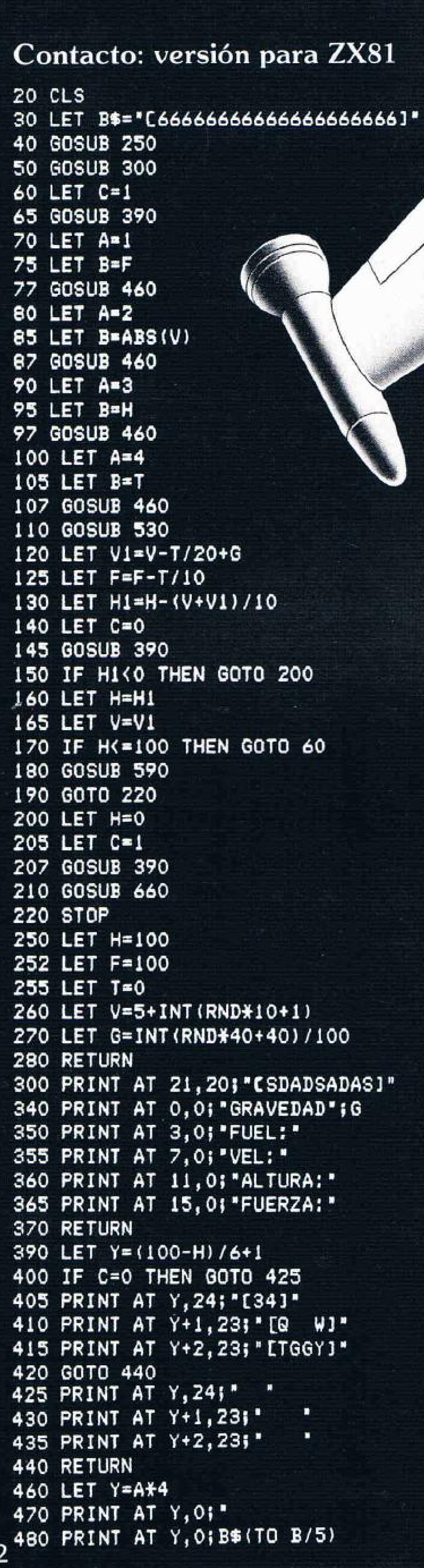
```
20 PRINT CHR$(147)CHR$(5);
25 POKE 36879,8
30 DEF FNR(X)=INT(RND(1)*X+1)
40 GOSUB 250
50 GOSUB 300
60 C=1: GOSUB 390
70 A=1:B=F: GOSUB 460
80 A=2:B=ABS(V): GOSUB 460
90 A=3:B=H: GOSUB 460
100 A=4:B=T: GOSUB 460
110 GOSUB 530
120 V1=V-T/20+G:F=F-T/10
130 H1=H-(V+V1)/10
140 C=0: GOSUB 530
150 IF H1<0 THEN 200
160 H=H1:V=V1
170 IF H<=100 THEN 60
180 GOSUB 590
190 GOTO 220
200 H=0:C=1: GOSUB 390
210 GOSUB 660
220 END
250 H=100:F=100:T=0
260 V=5+FNR(10)
270 G=(FNR(40)+40)/100
280 RETURN
300 FOR X=8178 TO 8185
320 POKE X,98+2*FNR(3)
330 NEXT
340 PRINT "GRAVEDAD=";G
350 PRINT "]]FUEL:"
355 PRINT "]]]VEL:"
360 PRINT "]]]ALTURA:"
365 PRINT "]]]FUERZA:"
370 RETURN
390 Y=8137-22*INT(H/5)
400 IF C=0 THEN 425
405 POKE Y,108: POKE Y+1,123
410 POKE Y+22,160: POKE Y+23,160
415 POKE Y+44,75: POKE Y+45,74
420 GOTO 440
425 FOR Z=0 TO 44 STEP 22
430 POKE Y+Z,32: POKE Y+Z+1,32
435 NEXT
440 RETURN
460 FOR X=0 TO 9
470 Y=A*88+X+7724
480 IF X<B/10 THEN POKE Y,102: GOTO 500
485 IF X<B/10+.5 THEN POKE Y,92: GOTO 500
490 POKE Y,32
500 NEXT
510 RETURN
530 GET I$
540 IF I$="]" THEN T=T+4: IF T>100 THEN T=100
550 IF I$="@" THEN T=T-4: IF T<0 THEN T=0
560 IF T>F THEN T=F
570 RETURN
590 PRINT CHR$(147)
600 FOR I=1 TO 20
610 POKE 7679+FNR(506),42
620 NEXT
630 PRINT "PERDIDOS EN EL ESPACIO!!"
640 RETURN
650 PRINT CHR$(147)"ATERRIZAJE"
660 PRINT "A LA VELOCIDAD"; INT((V+V1)*5)/10
670 IF (V+V1)<8 THEN PRINT "TODOS A SALVO": RETURN
680 PRINT "TODOS MUERTOS": RETURN
```

] es la tecla de cursor hacia abajo.

@ es la tecla de cursor hacia la derecha.



Contacto: versión para ZX81



```
20 CLS
30 LET B$="[66666666666666666666]"
40 GOSUB 250
50 GOSUB 300
60 LET C=1
65 GOSUB 390
70 LET A=1
75 LET B=F
77 GOSUB 460
80 LET A=2
85 LET B=ABS(V)
87 GOSUB 460
90 LET A=3
95 LET B=H
97 GOSUB 460
100 LET A=4
105 LET B=T
107 GOSUB 460
110 GOSUB 530
120 LET V1=V-T/20+G
125 LET F=F-T/10
130 LET H1=H-(V+V1)/10
140 LET C=0
145 GOSUB 390
150 IF H1<0 THEN GOTO 200
160 LET H=H1
165 LET V=V1
170 IF H<=100 THEN GOTO 60
180 GOSUB 590
190 GOTO 220
200 LET H=0
205 LET C=1
207 GOSUB 390
210 GOSUB 660
220 STOP
250 LET H=100
252 LET F=100
255 LET T=0
260 LET V=5+INT(RND*10+1)
270 LET G=INT(RND*40+40)/100
280 RETURN
300 PRINT AT 21,20;"[SDADSADAS]"
340 PRINT AT 0,0;"GRAVEDAD";G
350 PRINT AT 3,0;"FUEL:"
355 PRINT AT 7,0;"VEL:"
360 PRINT AT 11,0;"ALTURA:"
365 PRINT AT 15,0;"FUERZA:"
370 RETURN
390 LET Y=(100-H)/6+1
400 IF C=0 THEN GOTO 425
405 PRINT AT Y,24;"[34]"
410 PRINT AT Y+1,23;"[Q  W]"
415 PRINT AT Y+2,23;"[TGGY]"
420 GOTO 440
425 PRINT AT Y,24;"  "
430 PRINT AT Y+1,23;"    "
435 PRINT AT Y+2,23;"    "
440 RETURN
460 LET Y=A*4
470 PRINT AT Y,0;"
480 PRINT AT Y,0;B$(TO B/5)
510 RETURN
530 LET I$=INKEY$
540 IF I$="A" THEN LET T=T+4
545 IF T>100 THEN LET T=100
550 IF I$="D" THEN LET T=T-4
555 IF T<0 THEN LET T=0
560 IF T>F THEN LET T=F
570 RETURN
590 CLS
600 FOR I=1 TO 20
610 PRINT AT INT(RND*22),INT(RND*32);"*"
620 NEXT I
630 PRINT AT 10,10;"PERDIDOS EN EL ESPACIO"
640 RETURN
650 CLS
660 PRINT AT 20,0;"ATERRIZAJE A VELOCIDAD ";
    INT((V+V1)*5)/10
670 IF (V+V1)<8 THEN PRINT "A SALVO"
675 IF (V+V1)>=8 THEN PRINT "TODOS MUERTOS"
680 RETURN
```

Nota especial para los usuarios del ZX81

Los caracteres gráficos se muestran en el listado usando la siguiente convención: un '[' significa que se pulsa la tecla GRAPHICS, seguida de las letras y números a la vez que la tecla SHIFT, hasta que aparece ']'; en ese momento hay que pulsar la tecla GRAPHICS de nuevo, para volver al modo normal.



Contacto: versión para ZX Spectrum

```
20 CLS
30 DEF FNR(X)=INT(RND*X+1)
40 GOSUB 250
50 GOSUB 300
60 LET C=0: GOSUB 390
70 LET A=1: LET B=F: LET C=2*(F<25)
75 GOSUB 460
80 LET A=2: LET B=ABS V: LET C=4*(V<0)
85 GOSUB 460
90 LET A=3: LET B=H: LET C=2*(H<25)
95 GOSUB 460
100 LET A=4: LET B=T: LET C=0
105 GOSUB 460
110 GOSUB 530
120 LET V1=V-T/20+G: LET F=F-T/10
130 LET H1=H-(V+V1)/10
140 LET C=1: GOSUB 390
150 IF H1<0 THEN GOTO 200
160 LET H=H1: LET V=V1
170 IF H<=100 THEN GOTO 60
180 GOSUB 590
190 GOTO 220
200 LET H=0: LET C=0: GOSUB 390
210 GOSUB 650
220 STOP
250 LET H=100: LET F=100: LET T=0
260 LET V=5+FNR(10)
270 LET G=(FNR(40)+40)/100
280 RETURN
300 PLOT 180,8
310 FOR X=1 TO 15
320 DRAW 5,FNR(3)-2
330 NEXT X
340 PRINT "GRAVEDAD=";G
350 PRINT ''"FUEL:"''''"VEL:"
360 PRINT '''"ALTURA:"''''"FUERZA:"
370 RETURN
390 INVERSE C
400 LET Y=H*1.3+10
410 PLOT 200,Y: DRAW 34,0
420 DRAW -4,20: DRAW -13,10
430 DRAW -13,-10: DRAW -4,-20
440 RETURN
460 LET Y=172-A*32
470 INK C
480 PLOT 0,Y
490 DRAW B,0
500 DRAW INVERSE 1,100-B,0
510 RETURN
530 LET I$=INKEY$
540 IF I$="A" THEN LET T=T+4:
    IF T>100 THEN LET T=100
550 IF I$="D" THEN LET T=T-4:
    IF T<0 THEN LET T=0
560 IF T>F THEN LET T=F
570 RETURN
590 CLS
600 FOR I=1 TO 20
610 PRINT AT FNR(21),FNR(31);"*"
620 NEXT I
630 PRINT "PERDIDOS EN EL ESPACIO!!"
640 RETURN
650 PRINT AT 0,0;"VELOCIDAD DE CONTACTO ";
    INT((V+V1)*5)/10'
660 IF (V+V1)<8 THEN GOTO 680
670 PRINT "TODOS MUERTOS": RETURN
680 PRINT "TODOS A SALVO": RETURN
```

Contacto: versión para MSX

```
5 PRINT "INTRODUCE UN NUMERO POSITIVO"
7 INPUT Z
8 Z=RND(-Z)
10 CLS
20 SCREEN 2,1
30 DEF FNR(X)=INT(RND(1)*X+1)
40 A$=""
50 FOR I=1 TO 8
60 READ A
70 A$=A$+CHR$(A)
80 NEXT I
90 SPRITE$(1)=A$
100 GOSUB 300
120 GOSUB 350
130 PUT SPRITE 1,(124,(100-H)*1.3),15
150 GOSUB 530
160 V1=V-T/20+G:F=F-T/10
170 H1=H-(V+V1)/10
180 IF H1<0 THEN GOTO 660
190 H=H1:V=V1
200 IF<=100 THEN GOTO 130
210 GOTO 590
220 END
230 DATA 60,126,255,255,255,66,66,231
300 H=100:F=300:T=0
310 V=5+FNR(10)
320 G=(FNR(40)+40)/100
330 RETURN
350 X0=0:Y0=155
360 FOR X=0 TO 255 STEP 5
370 XN=X:YN=159-FNR(10)
380 LINE (X0,Y0)-(XN,YN)
390 X0=XN:Y0=YN
400 NEXT X
410 FOR I=1 TO 30:PSET(FNR(255),FNR(150)),15
420 NEXT I
430 RETURN
530 I$=INKEY$
540 IF I$="A" THEN T=T+4:IF T>100 THEN T=100
550 IF I$="D" THEN T=T-4:IF T<0 THEN T=0
560 IF T>F THEN T=F
570 RETURN
590 SCREEN 0:CLS
600 PRINT "PERDIDOS EN EL ESPACIO"
610 END
660 SCREEN 0
670 CLS
680 PRINT "ATERRIZAJE A VELOCIDAD=";
    INT((V+V1)*5)/10
690 IF (V+V1)<8 THEN 710
700 PRINT "TODOS MUERTOS":END
710 PRINT "TODOS A SALVO":END
```

Contacto: versión para Apple

```
15 HOME
20 HGR
30 DEF FNR(X)=INT(RND(1)*X+1)
40 GOSUB 250
50 GOSUB 300
60 C=3: GOSUB 390
70 A=1:B=F: GOSUB 460
80 A=2:B=ABS(V): GOSUB 460
90 A=3:B=H: GOSUB 460
100 A=4:B=T: GOSUB 460
110 GOSUB 530
120 V1=V-T/20+G:F=F-T/10
130 H1=H-(V+V1)/10
140 C=0: GOSUB 390
150 IF H1<0 THEN 200
160 H=H1:V=V1
170 IF H<=100 THEN 60
180 GOSUB 590
190 GOTO 220
200 H=0:C=3: GOSUB 390
210 GOSUB 660
220 END
250 H=100:F=100:T=0
260 V=5+FNR(10)
270 G=(FNR(40)+40)/100
280 RETURN
300 HCOLOR=3
305 HPLOT 0,155
310 FOR X=0 TO 279 STEP 5
320 HPLOT TO X,159-FNR(10)
330 NEXT
335 FOR I=1 TO 30: HPLOT FNR(279),FNR(150)
337 NEXT
340 VTAB 21: PRINT TAB(34);"G=";G
350 VTAB 21: PRINT "FUEL:": PRINT "VEL:"
360 PRINT "ALTURA:": PRINT "FUERZA:";
370 RETURN
390 HCOLOR=C
400 Y=(100-H)*1.3
410 HPLOT 140,Y TO 120,Y+10
420 HPLOT TO 120,Y+20: HPLOT TO 160,Y+20
430 HPLOT TO 160,Y+10: HPLOT TO 140,Y
435 HPLOT 155,Y+20 TO 160,Y+25
437 HPLOT 125,Y+20 TO 120,Y+25
440 RETURN
460 VTAB(20+A): HTAB 8
470 INVERSE
480 PRINT SPC(B/4);
490 NORMAL
500 PRINT SPC(26-B/4);
510 RETURN
530 I$="": IF PEEK(-16384)>27 THEN GET I$
540 IF I$="A" THEN T=T+4: IF T>100 THEN T=100
550 IF I$="D" THEN T=T-4: IF T<0 THEN T=0
560 IF T>F THEN T=F
570 RETURN
590 HOME : VTAB 23
600 PRINT "PERDIDO EN EL ESPACIO!!"
640 RETURN
660 HOME : VTAB 22
670 PRINT "ATERRIZAJE A VELOCIDAD ";
    INT((V+V1)*5)/10
680 IF (V+V1)<8 THEN 700
690 PRINT "TODOS MUERTOS": RETURN
700 PRINT "A SALVO": RETURN
```

Cómo ampliar los programas

Aquí damos algunas ideas que puedes añadir a los programas que incluimos en este libro o a tus propios programas. En muchos casos no podrás añadirlos al ZX81 con 1 K de memoria, ya que los juegos ocupan casi todo el espacio disponible.

Recuerda también que tendrás que limitar tus añadidos según los números de líneas disponibles en el programa o, de otro modo, tendrás que renumerarlo todo. Si decides renumerarlo, ten cuidado al cambiar todas las líneas GOTO y GOSUB.

Cómo hacer que el ordenador te diga cómo tienes que jugar

Puedes añadir una sección a cualquier programa para que el ordenador imprima las instrucciones del juego. El método más fácil para hacer esto es añadir algunas líneas, como las que damos a continuación, al comienzo del programa, y después poner una subrutina al final.

```
10 PRINT "TITULO DEL JUEGO"
11 PRINT "QUIERES SABER COMO"
12 PRINT "SE JUEGA?"
15 INPUT I$
■✱17 IF I$(1)="S" THEN GOSUB 1000
★▲●17 IF LEFT$(I$,1)="S" THEN GOSUB 1000
```

El programa principal queda así:

```
1000 PRINT "LO QUE TIENES QUE"
1010 PRINT "HACER ES ..."
1999 RETURN
```

Puedes añadir tantas sentencias de impresión como quieras para las instrucciones, pero recuerda poner un número y la palabra PRINT al comienzo de cada una. Limita la longitud de la frase que va entre comillas al número de caracteres que tu ordenador pueda imprimir en una línea. No olvides poner una línea RETURN al final o el programa no funcionará.

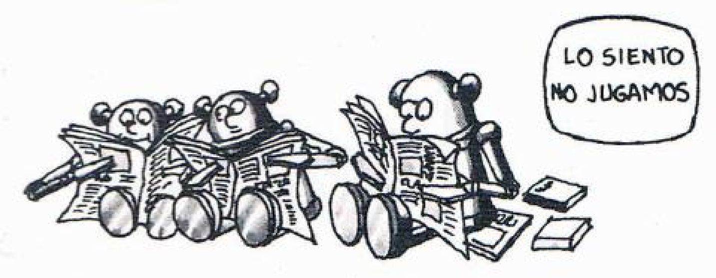
Cómo hacer que el ordenador se pare y espere



Si tus instrucciones son demasiado largas, debes insertar esta subrutina para que el programa se pare en un punto determinado, hasta que tú pulses una tecla. Esto te permite parar las instrucciones cuando van a desaparecer por la parte superior de la pantalla. Pon una línea GOSUB en el sitio que quieres que se pare el programa y esta subrutina al final:

```
1000 PRINT "PULSA UNA TECLA PARA CONTINUAR ";
★■✱1010 IF INKEY$="" THEN GOTO 1010
●1010 GET I$
▲1010 GET I$:IF I$="" THEN GOTO 1010
1020 PRINT
1030 RETURN
```

Cómo hacer que el ordenador te hable



Puedes hacer que el ordenador te haga preguntas y reaccione a tus respuestas. Por ejemplo, estas líneas harán que el ordenador rehúse jugar contigo a menos que tu nombre comience por J:

```
1 PRINT "CUAL ES TU NOMBRE?"
2 INPUT I$
3 IF I$(1)<>"J" THEN GOTO 1000
3 IF LEFT$(I$,1)<>"J" THEN GOTO 1000
4 PRINT "DE ACUERDO, PUEDES JUGAR"
5 PRINT "ESTAS LISTO?"
6 INPUT J$
■✱7 IF J$(1)<>"S" THEN GOTO 5
★▲●7 IF LEFT$(J$,1)<>"S" THEN GOTO 5
```

El programa principal queda así:

```
1000 PRINT "LO SIENTO, ESTE JUEGO ES"
1010 PRINT "SOLO PARA LA GENTE"
1020 PRINT "CUYOS NOMBRES EMPIEZAN"
1030 PRINT "CON J"
```

En este otro ejemplo, te desafía a ser lo suficientemente valiente como para atreverte a jugar.

```
   10 PRINT "JUEGO MUY PELIGROSO"
   12 PRINT "ERES LO BASTANTE VALIENTE"
   14 PRINT "PARA PEGAR AL MONSTRUO"
   15 PRINT "DE LOS PELOS VERDES?"
   16 INPUT I$
 ✱■17 IF I$(1)="S" THEN GOTO 20
★▲●17 IF LEFT$(I$,1)="S" THEN GOTO 20
   18 PRINT "COBARDICA"
   19 STOP
```

Puedes combinar este listado con la subrutina de instrucciones, tomando las líneas 11 a 17 de la sección de instrucciones que damos en esta página y poniéndolas en las líneas 20 a 26 de este programa. Puedes comenzar el programa principal en la línea 30 y añadir la subrutina de instrucciones al final.

¿Quieres otra oportunidad?

En lugar de teclear RUN cada vez que quieras jugar de nuevo, puedes hacer que el ordenador te pregunte si deseas jugar de nuevo. Pon estas líneas al final del programa, justo antes de la última sentencia STOP:

```
   1000 PRINT "QUIERES JUGAR OTRA VEZ?"
   1010 INPUT I$
 ■✱1020 IF I$(1)="S" THEN RUN
★▲●1020 IF LEFT$(I$,1)="S" THEN RUN
   1030 PRINT "DE ACUERDO, ADIOS"
   1040 STOP
```

Cambia los números de línea de acuerdo con tu programa.



Cómo producir efectos de sonido

El VIC 20, el ZX Spectrum, los MSX y los Apple II permiten producir sonidos; puedes añadir líneas a tus programas para producirlos en los lugares apropiados. Podrás añadir, por ejemplo, una explosión o una pequeña melodía que suene si tú eres el ganador. Cada ordenador necesita diferentes instrucciones para producir sonidos, de modo que lo mejor es que consultes tu manual. En casos bastará añadir una línea al programa en el lugar en que tú quieras que suene. En otros, necesita diversas líneas y será mejor ponerlas como una subrutina. En la parte de atrás del manual del VIC, encontrarás algunas subrutinas muy útiles para hacer sonidos como "disparo de láser", "explosión" y "alerta roja". Pon una línea GOSUB donde quieras que aparezca el sonido, numera la subrutina y añade un RETURN al final de ella.

Nota para ordenadores MSX

Debido al especial sistema de generar números aleatorios que tienen los ordenadores MSX, es necesario que todos los programas para estas máquinas que utilicen la función RND(1) incluyan esta pequeña subrutina al principio:

```
10 PRINT "INTRODUCE UN NUMERO POSITIVO"
20 INPUT Z
30 Z=RND(-Z)
```

Debiéndose ejecutar antes que cualquier RND(1). Esto hace que el generador se inicialice generando una secuencia distinta, ya que si no se ponen estas líneas, siempre que ejecutemos el programa sacará la misma secuencia. El número que pedirá debe ser cualquiera superior a cero.

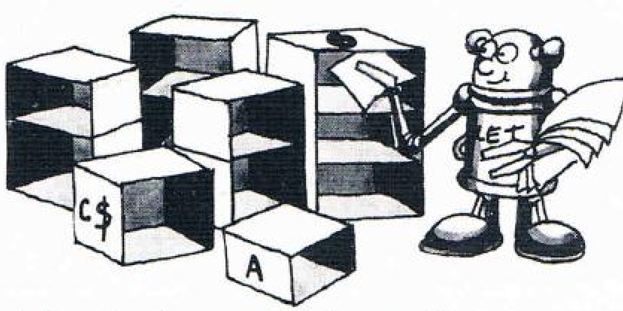
Nota especial para los usuarios del Spectrum

Si tienes un ZX Spectrum, encontrarás que algunos de los juegos de este libro funcionan demasiado rápido. Encontrarás al lado del programa un recuadro conteniendo las instrucciones para cambiar la velocidad.
Recuerda que para disminuir la velocidad del juego tienes que utilizar un número más alto.

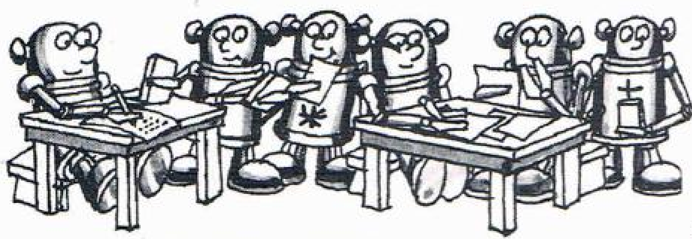
Cómo escribir tus propios programas

A medida que practiques con los juegos de este libro, probablemente querrás más y más cambios en ellos y, lógicamente, pensarás escribir nuevos juegos de tu propia cosecha. En estas dos páginas encontrarás algunas indicaciones sobre cómo hacerlo.

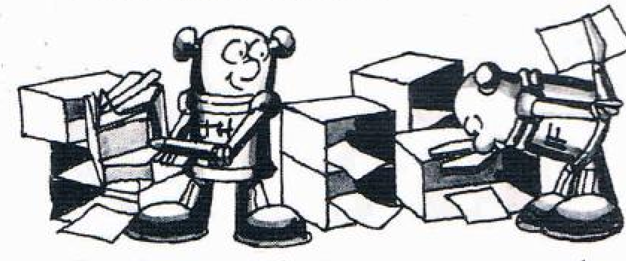
Antes de que empieces, es una buena idea pensar sobre lo que tu ordenador puede o no puede hacer.



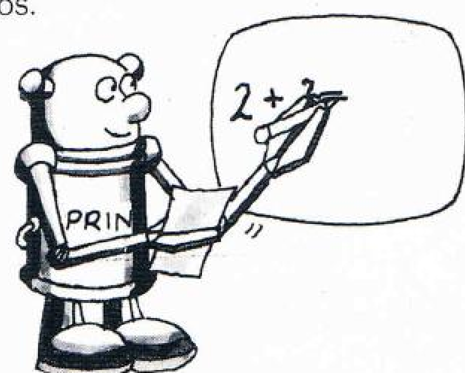
* Puede almacenar información.



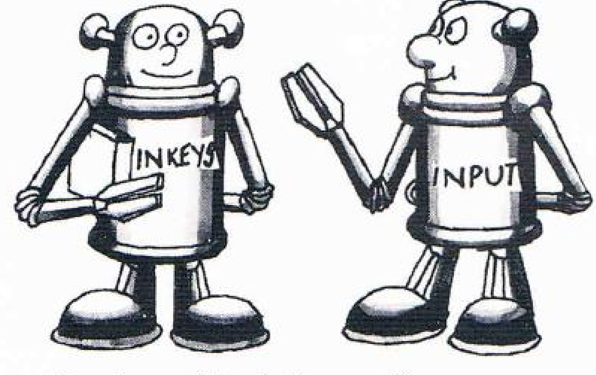
* Puede hacer cálculos.



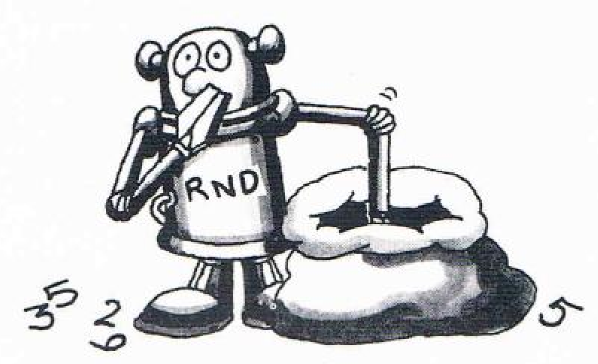
* Puede tomar decisiones comparando los elementos de información de varios modos.



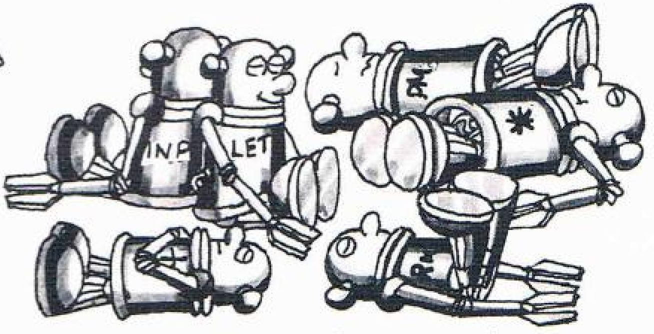
* Puede darte los resultados de sus cálculos y decisiones y también lo que está almacenado en su memoria.



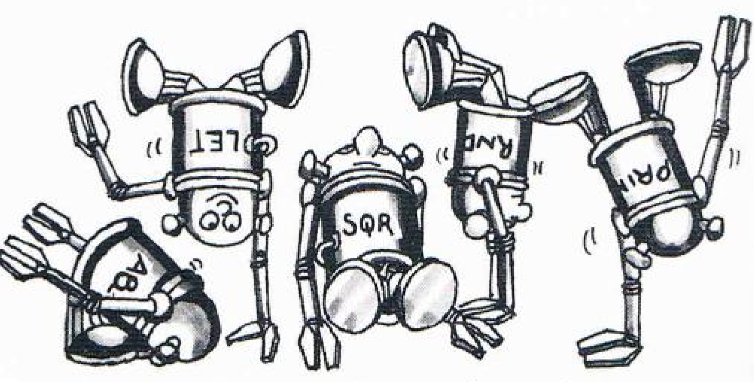
* Puede pedirte información.



* Puede seleccionar números aleatoriamente utilizando RND.



* No puede hacer nada si no se lo dices tú.



* Suponiendo que utilices su lenguaje correctamente, puede hacer exactamente sólo lo que le digas, incluso si es una estupidez.

Recuerda, cuando estés planeando un juego, no incluir nada que tu ordenador no pueda hacer.

Cómo planear un juego

Antes de que le digas al ordenador en qué consiste tu juego, debes saber exactamente cómo jugar y cuáles son las reglas. El ordenador necesitará un conjunto de sencillas instrucciones lógicas, de modo que haz funcionar el juego en tu cabeza o en un papel primero y divídelo en pasos simples.

A continuación, escribe un esquema (en español, no uses BASIC aún) de todas las etapas del juego en orden.

Aquí incluimos un esquema para un simple juego de disparar, cómo disparar balas de cañón a la nave pirata o disparar rayos láser a un invasor alienígena, para darte una idea.

PLAN

1) IMPRIMIR TÍTULOS E INSTRUCCIONES.

2) ELEGIR UN OBJETIVO PARA ESTE JUEGO.

3) EMPEZAR UN BUCLE QUE LE DA AL JUGADOR N INTENTOS.

4) ESPERAR A QUE EL JUGADOR DISPARE.

5) MIRAR SI EL TIRO DA EN EL BLANCO.

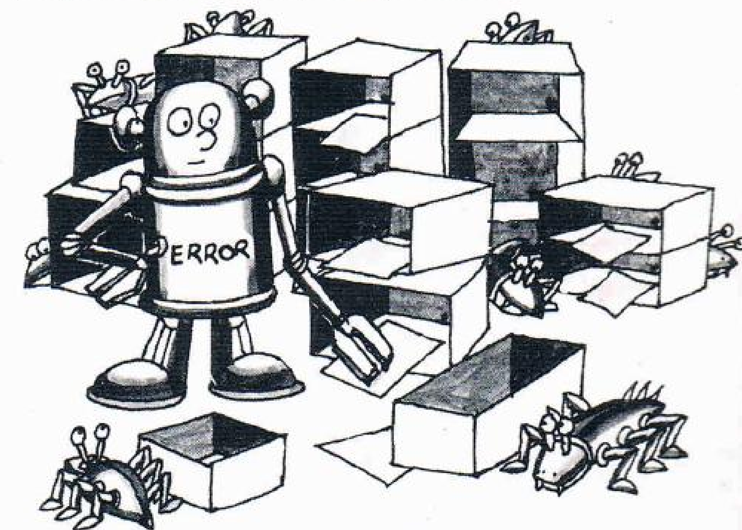
6) IMPRIMIR UN MENSAJE DEPENDIENDO DE LA PUNTERIA.

7) VOLVER POR OTRO INTENTO, SI NO TUVO ÉXITO.

Escritura del programa

La próxima etapa es traducir tu esquema a BASIC. Cada paso de tu esquema puede necesitar varias líneas en BASIC. No olvides dejar espacios cuando numeres las líneas del programa; así podrás volver atrás y añadir más si lo necesitas.

Haz primero un boceto del programa en un papel y después comienza a comprobar en el ordenador. Él descubrirá los errores con mayor rapidez y puede darte una pista sobre dónde está equivado el programa. Evitar o localizar errores en un programa es difícil incluso para expertos: no esperes que tus programas funcionen a la primera.

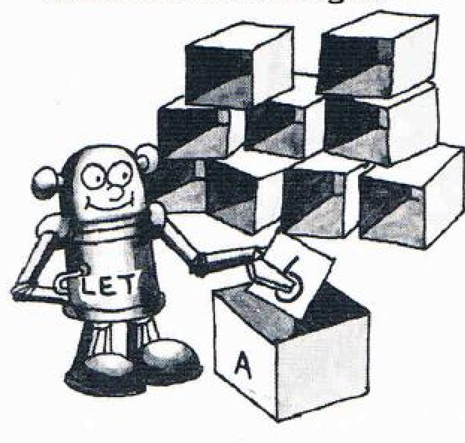


Una vez que hayas puesto en funcionamiento el meollo del programa, puedes añadir marcadores, comentarios extras, más objetivos, etc. Podrás incorporar secciones de los programas de este libro a tus juegos.

No esperes escribir juegos excitantes y originales rápidamente. Mantén tus ideas sencillas y prepárate para adaptarlas según vayas escribiendo el programa. Puedes descubrir que has incluido algo en tu juego que es fácil de hacer para los humanos, pero muy difícil para un ordenador. Cuando tengas más experiencia, sabrás instintivamente qué es lo que tu ordenador puede hacer y la manera más fácil de escribir el programa.

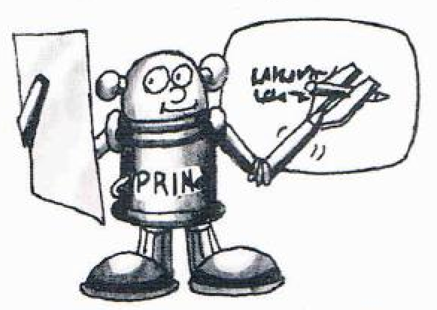
Sumario del BASIC

Esta sección te da algunas de las instrucciones frecuentes en BASIC, describe qué es lo que le hacen hacer al ordenador y cómo son usadas. Muchas de ellas han sido utilizadas en los programas de este libro, de modo que puedes volver atrás y comprobar cómo funcionan en un juego; pero no todas pueden ser usadas en todos los ordenadores mencionados aquí. La tabla de conversión de la página 46 muestra cuáles pueden ser utilizadas en su lugar.



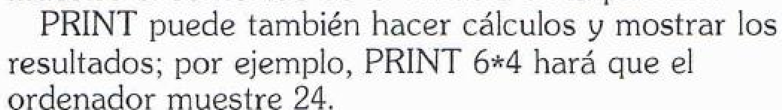
LET le dice al ordenador que reserve una sección de su memoria y ponga un valor particular en ella; por ejemplo, LET A=6 significa *reserva una sección de memoria "A" y pon el valor 6 en ella.* A "A" se la llama "variable" y poner algo en ella se llama "asignar un valor a una variable".

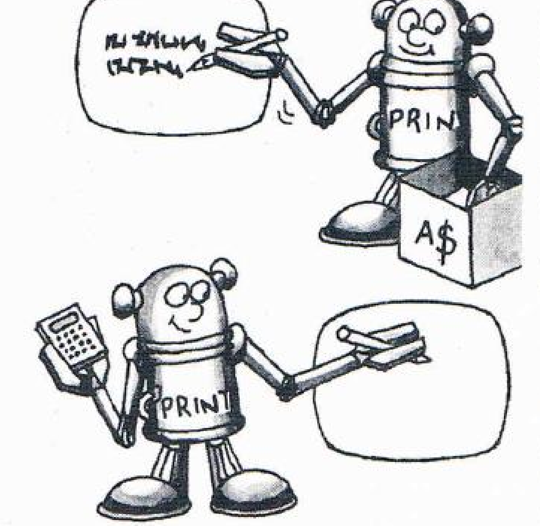
El nombre de algunas variables está seguido por un signo dólar; por ejemplo, A$. Esto significa que son "cadenas alfanuméricas", que pueden contener varios caracteres, incluyendo letras, números y símbolos.



PRINT dice al ordenador que muestre cosas en la pantalla, y lo puedes utilizar así:

Un mensaje encerrado entre comillas con PRINT delante de él, aparecerá en la pantalla tal como tú lo tecleaste. El trozo entre comillas no hace falta que esté en BASIC, puede ser lo que quieras. PRINT seguido del nombre de una variable, por ejemplo, PRINT A o PRINT A$, le dice al ordenador que muestre el contenido de la variable en la pantalla.





PRINT puede también hacer cálculos y mostrar los resultados; por ejemplo, PRINT 6*4 hará que el ordenador muestre 24.

Puedes usar PRINT, sin nada más, para dejar una línea en blanco.

RND dice al ordenador que elija un número aleatoriamente. Diferentes ordenadores utilizan diferentes formas de RND y puedes verlas en la tabla de conversión de caracteres de la página 46. En los ordenadores Sinclair, RND, sin nada más, produce un número entre 0 y 0,99999999. Puedes variar los límites del número que quieres que genere, multiplicando RND y añadiéndole otro. Por ejemplo, RND*20 produce un número entre 0 y 19,99999999, mientras que RND*20+1 produce un número entre 1 y 20,99999999.

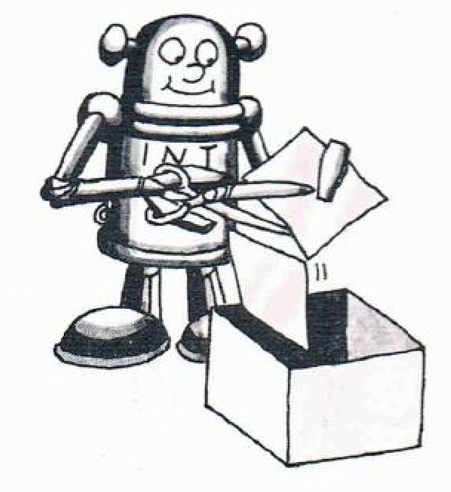
Ver INT para saber cómo producir sólo números enteros.

Ver CHR$ para saber cómo producir letras y otros caracteres del teclado aleatoriamente.



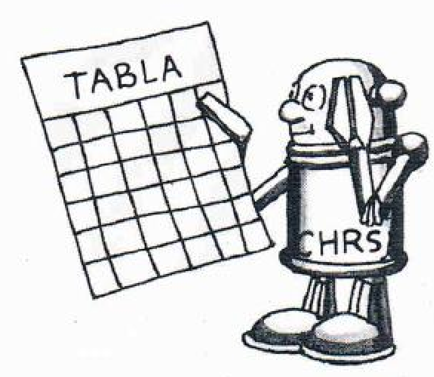
INT es una abreviatura de *integer* (en español, enteros), que significa *números enteros*. Para números positivos, le dice al ordenador que ignore todo lo que está a la derecha del punto decimal; por ejemplo, INT(20,999) es 20. Para números negativos que ignore todo lo que esté a la derecha del punto decimal e "incremente" el número a la izquierda por uno; por ejemplo, INT(-3,6) es -4.

El INT es bastante usado con RND, como INT(RND*20+1) que le dice al ordenador que quieres elegir un número entre 1 y 20.



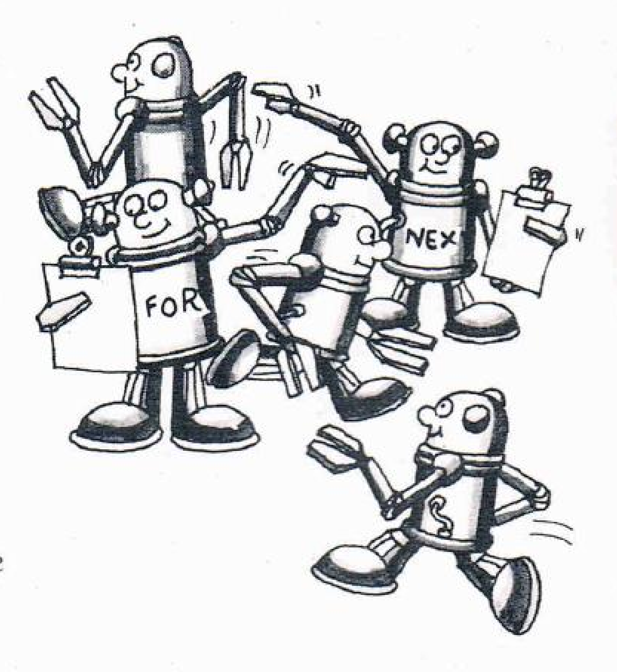
CHR$ convierte números en letras. Excepto el ZX81, todos los ordenadores de este libro utilizan el juego de caracteres ASCII* del teclado en el que cada carácter se corresponde a un número fijo. Por ejemplo, la letra A tiene el código numérico 65 y PRINT CHR$(65) mostrará una A en la pantalla.

Puedes usar CHR$ con INT y RND para hacer que el ordenador seleccione letras aleatoriamente, por ejemplo:
CHR$(INT(RND*26+65))
Esta línea producirá letras aleatoriamente en un ZX Spectrum (ver tabla de conversión para otros ordenadores).



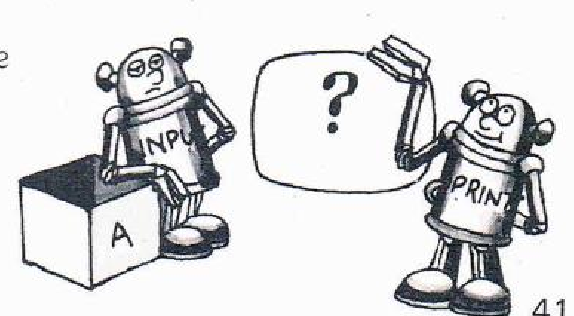
FOR es usado para comenzar un "bucle" que hará que el ordenador repita una parte del programa un número fijo de veces. Debe ir seguido de una variable (como, por ejemplo, G, para ver el número de intentos que permite el juego) y se debe dar un valor inicial y un valor final a la variable.

El fin del bucle está marcado por una línea NEXT (NEXT G, en este ejemplo), que incrementa el valor de la variable en 1 cada vez y, entonces, envía al ordenador de vuelta a la línea FOR de nuevo. Cuando la variable alcanza el valor final, el ordenador ignora la línea NEXT y ejecuta la línea siguiente. Todos los FOR deben tener un NEXT o se producirá un error.

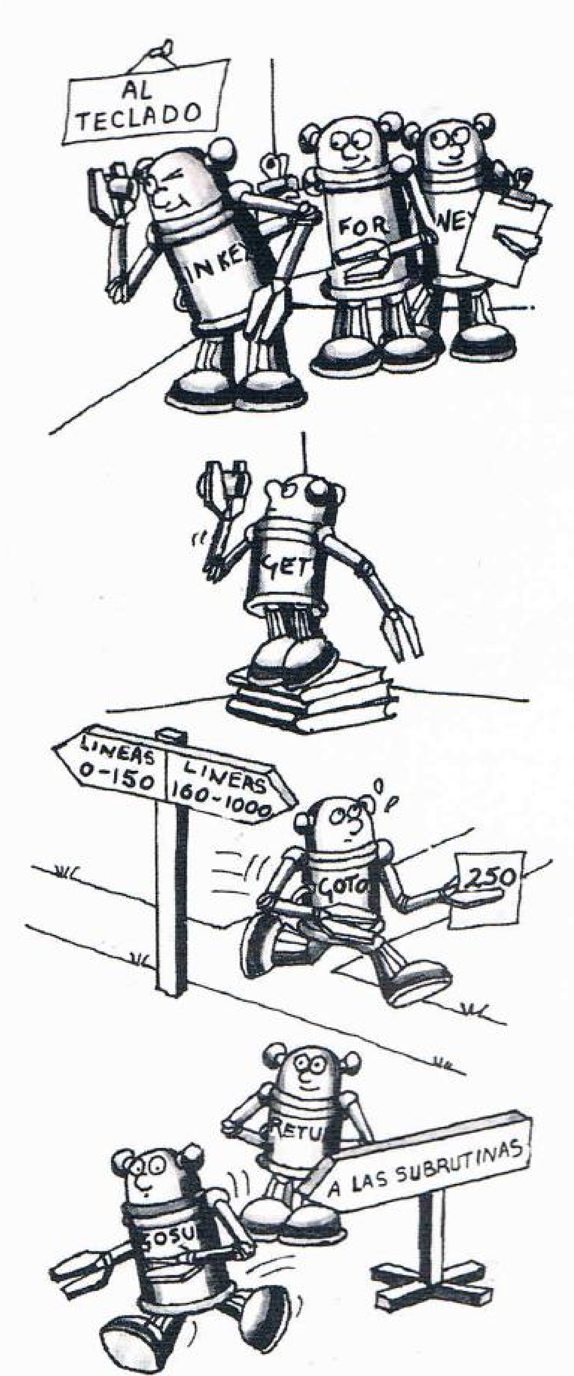


INPUT reserva un espacio en la memoria del ordenador, imprime una interrogación y espera a que teclees alguna cosa, que pondrá en ese espacio de memoria. No podrá seguir ejecutando el resto del programa hasta que pulses RETURN o ENTER.

Puedes usar variables numéricas o alfanuméricas con INPUT, pero si usas una variable numérica el ordenador no aceptará que teclees letras.



* American Standar Code for Information Interchange (Código Americano Estándar para Intercambio de Información).



INKEY$ comprueba el teclado para ver si una tecla ha sido pulsada y cuál es. No espera a que pulses una tecla, como hace el INPUT. Esta instrucción se usa normalmente en un bucle en el que el ordenador mira el teclado varias veces. Esto es debido a que los ordenadores funcionan tan rápido, que no tendrías oportunidad de pulsar una tecla en el tiempo que tarda el ordenador en hacer una comprobación.

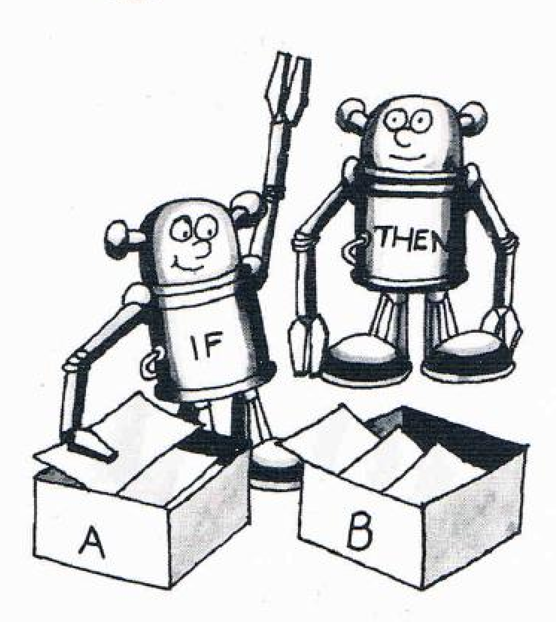
Si no has pulsado una tecla antes de que el bucle finalice, el ordenador seguirá con una cadena que no contiene nada (llamada una cadena "nula").

Nota: El Apple y el VIC no utilizan INKEY$.

GET es usada en lugar de INKEY$ en el VIC 20.

GOTO hace que el ordenador salte hacia arriba o abajo del programa ignorando las líneas de en medio. Debes poner el número de la línea a la que quieres saltar después de la instrucción GOTO.

GOSUB le dice al ordenador que deje el programa principal y vaya a la subrutina. GOSUB debe ir seguido del número de la primera línea de una subrutina. Al final de la subrutina debes tener una línea RETURN. Ésta envía al ordenador de vuelta al programa principal, a la línea que sigue inmediatamente después a la línea GOSUB. Un GOSUB sin un RETURN en un programa dará un error.

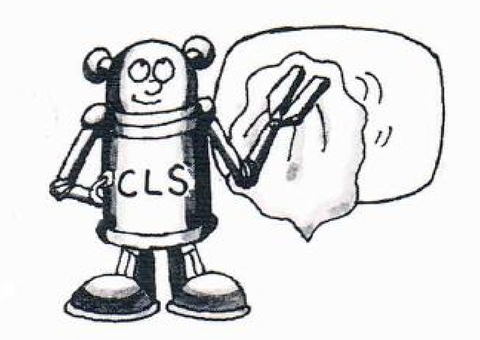


IF...THEN dice al ordenador que decida si una expresión es verdadera o falsa y que haga diferentes cosas, dependiendo de la respuesta. Es usada con AND u OR y también con los siguientes signos:

- = igual que
- < menor que
- > mayor que
- <= menor o igual que
- >= mayor o igual que
- <> distinto de

Si el ordenador decide que una expresión es verdadera, realiza la acción de la instrucción que sigue al THEN. Si decide que es falsa, entonces ignora el resto de la línea y va a la siguiente.

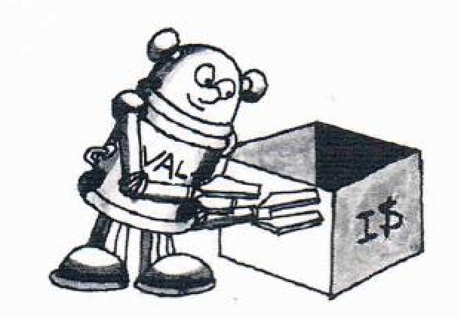
CLS se usa para borrar cualquier cosa que esté en la pantalla, sin borrar o cambiar lo que esté en memoria. Es útil para quitar el listado de la pantalla al ejecutar el programa o, en juegos, cuando quieres que el jugador reaccione al ver algunas cosas en un tiempo limitado. (Nota: El Apple y el VIC no utilizan CLS —ver tabla de conversión.)



HOME es usada por los ordenadores Apple en lugar de CLS para limpiar la pantalla.

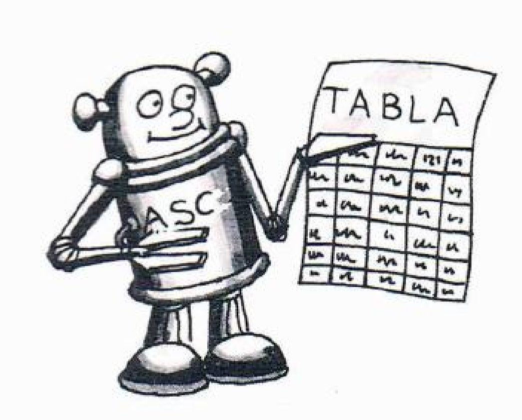
ABS ignora los signos más y menos delante de los números y toma sus valores "absolutos". Por ejemplo, ABS (-10) es 10 y ABS (+10) es también 10.

VAL toma el valor numérico de números escritos como cadenas alfanuméricas. En efecto, le dice al ordenador que ignore el signo dólar y trate a la cadena como una variable numérica normal. ejemplo, si I$="60", entonces VAL(I$) es el número 60.



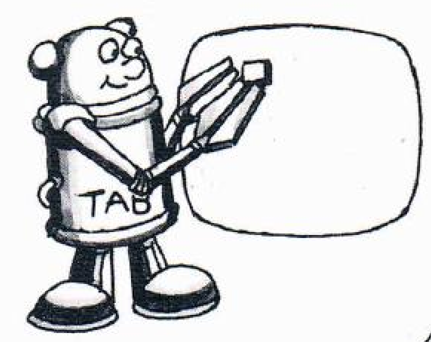
ASC convierte un carácter en su código ASCII; por ejemplo, ASC("3") da 51. La expresión entre paréntesis debe ser una cadena, por ejemplo, ASC(A$) o ASC("20").

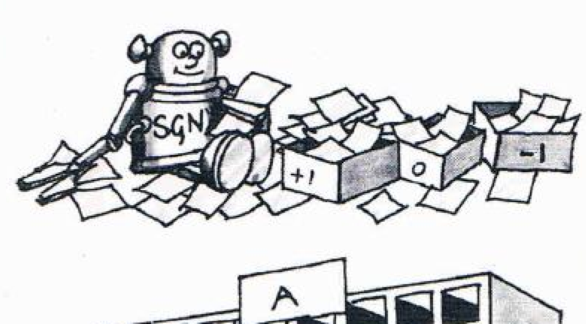
Nota: el ZX81 y el ZX Spectrum no utilizan ASC, aunque el Spectrum utiliza el código ASCII.



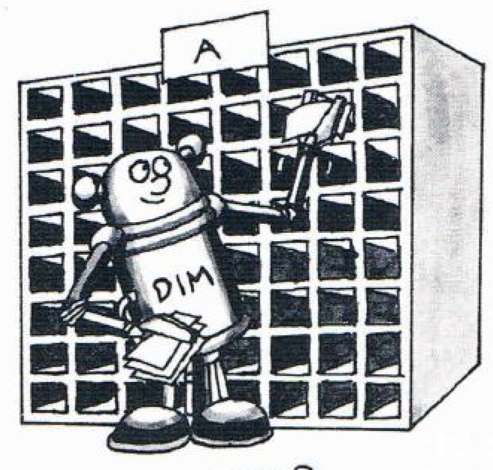
CODE es usada por el ZX81 y Spectrum en lugar de ASC. Al igual que ASC, debe estar seguido por una cadena. Recuerda que el ZX81 utiliza números de código diferentes de los de los demás ordenadores.

TAB mueve el cursor por la pantalla al número de columna que se le indique. Es usado normalmente con PRINT para mostrar algunas cosas en medio de la pantalla. El número de espacios que quieres que el cursor se mueva se pone entre paréntesis después de TAB. El número máximo que puedes usar depende de la anchura de la pantalla de tu ordenador.

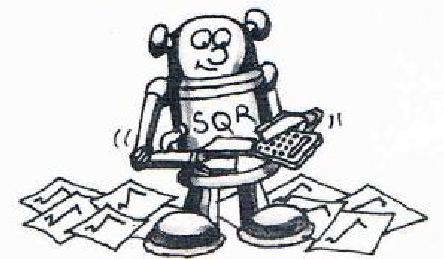




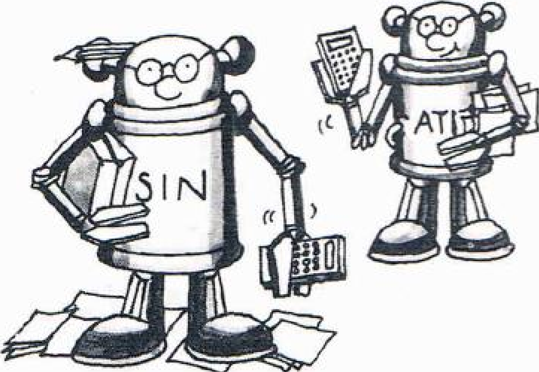
SGN dice al ordenador que encuentre el signo de un número. Produce -1 para un número negativo, 0 para cero y +1 para números positivos. Por ejemplo, SGN(-30) es -1, SGN(7) es +1 y SGN(0) es 0.



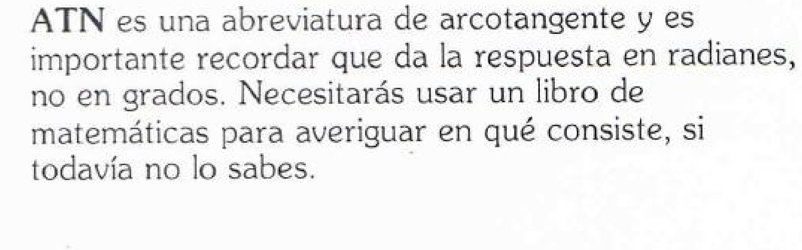
DIM dice al ordenador el espacio de memoria que será necesario para una "matriz" (una columna o una cuadrícula). Por ejemplo, DIM X(6) dice al ordenador que reserve un área suficientemente grande para contener una fila de 6 elementos, llamada X. DIM A(8,8) significa un espacio de memoria llamado A y lo suficientemente grande como para contener 8 elementos a lo largo y 8 hacia abajo. El número de elementos de datos usado en el programa debe corresponder al número de paréntesis después de DIM, o producirá un error.



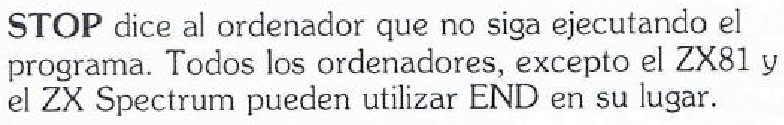
SQR hace raíces cuadradas de números. Por ejemplo, SQR(16) da como resultado 4.



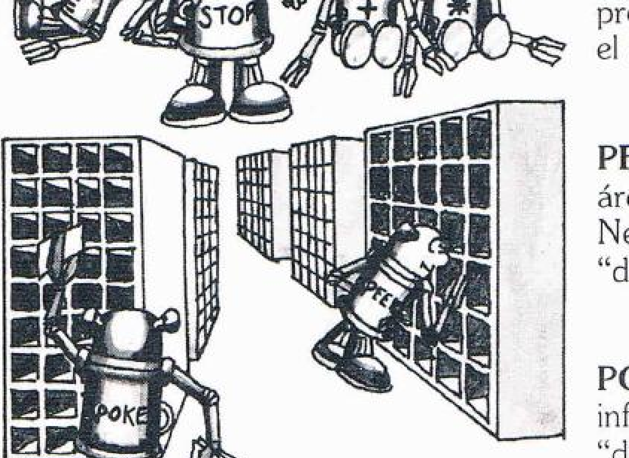
SIN calcula el seno de un ángulo. En un triángulo rectángulo, la longitud del lado opuesto a un ángulo, dividida por la longitud de la hipotenusa (el lado opuesto al ángulo recto) es el seno del ángulo. Cuando utilices SIN en un programa, el ángulo que estés usando debe estar medido en radianes, no en grados.



ATN es una abreviatura de arcotangente y es importante recordar que da la respuesta en radianes, no en grados. Necesitarás usar un libro de matemáticas para averiguar en qué consiste, si todavía no lo sabes.



STOP dice al ordenador que no siga ejecutando el programa. Todos los ordenadores, excepto el ZX81 y el ZX Spectrum pueden utilizar END en su lugar.



PEEK es un método para averiguar qué hay en un área específica de la memoria del ordenador. Necesitas usarlo con un número que especifique la "dirección" de la memoria.

POKE es un método especial para poner información en la memoria del ordenador usando una "dirección" de la memoria.

Tabla ASCII

Número de código	Carácter ASCII	Número de código	Carácter ASCII
32	espacio	62	>
33	!	63	?
34	"	64	@
35	#	65	A
36	$	66	B
37	%	67	C
38	&	68	D
39	'	69	E
40	(	70	F
41	)	71	G
42	*	72	H
43	+	73	I
44	,	74	J
45	-	75	K
46	.	76	L
47	/	77	M
48	0	78	N
49	1	79	O
50	2	80	P
51	3	81	Q
52	4	82	R
53	5	83	S
54	6	84	T
55	7	85	U
56	8	86	V
57	9	87	W
58	:	88	X
59	;	89	Y
60	<	90	Z
61	=		

Tabla de códigos del ZX81

Número de código	Carácter del ZX81	Número de código	Carácter del ZX81
11	"	41	D
12	£	42	E
13	$	43	F
14	:	44	G
15	?	45	H
16	(	46	I
17	)	47	J
18	>	48	K
19	<	49	L
20	=	50	M
21	+	51	N
22	-	52	O
23	*	53	P
24	/	54	Q
25	;	55	R
26	,	56	S
27	.	57	T
28	0	58	U
29	1	59	V
30	2	60	W
31	3	61	X
32	4	62	Y
33	5	63	Z
34	6		
35	7		
36	8		
37	9		
38	A		
39	B		
40	C		

Tabla de tamaños de pantalla

	Máximo número de caracteres a lo ancho (o número de columnas)	Máximo número de líneas de altura (o número de líneas)
VIC 20	22	23
ZX81	32	22
ZX Spectrum	32	22
Apple	40	24
MSX	40	24
Commodore 64	40	24

Tabla de conversión

Esta guía de referencia muestra algunas de las variaciones del BASIC usadas por las máquinas empleadas en este libro. No incluye instrucciones de gráficos, sonido o color, porque éstos varían enormemente de una máquina a otra. Fíjate que, aunque todos los ordenadores tienen PEEK y POKE, no usan el mismo sistema de direcciones de memoria, de modo que los números usados con PEEK y POKE deben cambiarse para cada ordenador.

	MSX	VIC/C64	Apple	ZX Spectrum	ZX81
crea un número aleatorio entre 0 y 0.99999999	RND(1)	RND(1)	RND(1)	RND	RND
crea un número aleatorio entre 1 y N	RND(1)*N+1	RND(1)*N+1	RND(1)*N+1	RND*N+1	RND*N+1
crea una letra aleatoria entre A y Z	CHR$(INT(RND(1)*26+65))	CHR$(INT(RND(1)*26+65))	CHR$(INT(RND(1)*26+65))	CHR$(INT(RND*26+65))	CHR$(INT(RND*26+38))
borra la pantalla	CLS	PRINT CHR$(147)	HOME	CLS	CLS
mira si se ha pulsado una tecla	INKEY$	GET X$	X$=" " IF PEEK(-16384) > 127 THEN GET X$	INKEY$	INKEY$
convierte caracteres en códigos	ASC("X") (cód. ASCII)	ASC("X") (cód. ASCII)	ASC("X") (cód. ASCII)	CODE("X") (cód. ASCII)	CODE("X") (cód. ZX81)
mover cursor hacia arriba	PRINT CHR$(30)	PRINT CHR$(145)	CALL-998	PRINT CHR$(11)	PRINT CHR$(112)
mover cursor hacia abajo	PRINT CHR$(31)	PRINT CHR$(17)	PRINT CHR$(10)	PRINT CHR$(10)	PRINT CHR$(113)
mover cursor hacia la izquierda	PRINT CHR$(29)	PRINT CHR$(157)	PRINT CHR$(8)	PRINT CHR$(8)	PRINT CHR$(114)
mover cursor hacia la derecha	PRINT CHR$(28)	PRINT CHR$(29)	PRINT CHR$(21)	PRINT CHR$(9)	PRINT CHR$(115)
coge los primeros N caracteres de una cadena	LEFT$(A$,N)	LEFT$(A$,N)	LEFT$(A$,N)	A$(1 TO N)	A$(1 TO N)
coge los últimos N caracteres de una cadena	RIGHT$(A$,N)	RIGHT$(A$,N)	RIGHT$(A$,N)	A$(N TO)	A$(N TO)
coge los N caracteres de en medio de una cadena	MID$(A$,N1,N2)	MID$(A$,N1,N2)	MID$(A$,N1,N2)	A$(N1 TO N2)	A$(N1 TO N2)



Claves

Encontrarás que tus respuestas a algunos de los "Rincones de los problemas" son diferentes a las incluidas aquí. Si funcionan en tu ordenador, son válidas; pero comprueba si son tan claras y sencillas como las de este libro.

Página 5
Despegue espacial

Las líneas 30 y 40 seleccionan los números que determinan qué fuerza es necesaria. Para incrementar el rango de fuerzas posibles, puedes incrementar el 20 en la línea 30 o el 40 en la línea 40, o ambos números. Esto, obviamente, hará el juego más difícil.

Página 7
Juegos intergalácticos

Cambia las líneas 220 y 230 por las siguientes:

```
222 LET B=B+INT(1000/G)
230 GOTO 20
```

y añade una nueva línea 15:

```
15 LET B=0
```

Página 9
El malvado extraterrestre

Cambia las líneas 20 y 30 y añade una nueva línea 25, como sigue:

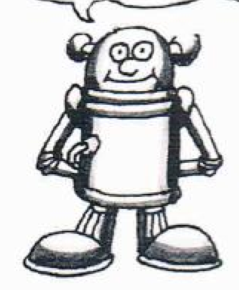
```
20 PRINT "DIFICULTAD? (6-30)"
25 INPUT S
30 LET G=INT(S/3)
```

Página 11
Golpea los ojos de la oruga

Para hacer que las orugas aparezcan en más de 4 sitios en la pantalla, tienes que poner un número mayor que 4 en la mitad de la línea 70, cambiar la línea 80 y añadir más subrutinas al final del programa (una por cada posición extra).

Aquí damos los cambios para hacer que las orugas aparezcan en 5 sitios:

```
    70 LET R=INT(RND*5+1)
■✱  80 GOSUB 220+20*R
★▲● 80 ON R GOSUB 240,260,280,300,320
    240 LET D=5
    245 LET A=1
    250 GOTO 350
    260 LET D=1
    265 LET A=9
    270 GOTO 350
    280 LET D=5
    285 LET A=18
    290 GOTO 350
    300 LET D=10
    305 LET A=7
    310 GOTO 350
    320 LET D=15
    325 LET A=15
    330 GOTO 350
```



Para añadir más orugas, cambia el 10 en las líneas 30 y 220 por un número mayor. (Asegúrate, que utilizas el mismo número para las dos líneas.)

Página 13
Alunizaje

Para incrementar la velocidad permitida para un aterrizaje seguro, necesitas hacer cambios en las líneas 230, 240 y 250. Puedes usar los números que quieras; cuanto más alto sea el número, más fácil será el juego. En este ejemplo se te permite una velocidad de 2 para un buen aterrizaje y de 7 para un aterrizaje "pasable":

```
230 IF V1>7 THEN PRINT "CHOCASTE-TODOS
    MUERTOS"
240 IF V1>2 AND V1<7 THEN PRINT "VALE,
    PERO ALGUNOS HERIDOS"
250 IF V1<=2 THEN PRINT "ATERRIZAJE
    PERFECTO"
```

Página 15
Los monstruos de Galacticón

Cuatro métodos para hacer que el juego sea más difícil:

1. Comenzar el juego con menos gente en el grupo, poniendo un número inferior a 5 en la línea 40.

2. Incrementar el número de monstruos, cambiando el 4 en las

líneas 20 y 30. Añade los nombres de los monstruos extras en las líneas 81 a 89, usando M$(5) y M$(6).

3. Reducir el número de intentos alterando el 8 en la línea 160.

4. Incrementar la posibilidad de que el monstruo sea dañado, aumentando el 4 en la línea 330.

Página 17
Cazas extraterrestres

En este juego, N es el número de código. Para cambiar la puntuación de modo que se ajuste al número de código, has de incrementar cada vez la puntuación sumándole N en vez de 1. Para ello, cambia la línea 190:

```
■✵190 IF I$=CHR$(CODE(L$)+N) THEN LET
      S=S+N
★▲●190 IF I$=CHR$(ASC(L$)=N) THEN LET
      S=S+N
```

Página 19
Cinturón de asteroides

Cambia la línea 260 para que el ordenador añada ese número de estrellas (asteroides) a tu marcador, en lugar de 1. El número de estrellas es controlado por el valor elegido por N en la línea 70, así que, como en el juego anterior, has de añadir N al marcador. También debes cambiar la línea 320.

```
260 LET S=S+N
320 PRINT "TU PUNTUACION ES ";S;
    " PUNTOS"
```

Página 21
Viaje al futuro

1. Para incrementar el número de años que deben transcurrir antes de que vuelvas a la Tierra, cambia el 100 en la línea 30 por un número mayor; por ejemplo, 150, quedando así:

```
30 LET T=INT(RND*150+25)
```

2. Para incrementar la precisión de 5 a 2 años, cambia los 5 en las líneas 180 y 190 por 2, quedando así:

```
180 IF ABS(T-T2)<=2 THEN PRINT
    "LLEGASTE A TIEMPO"
190 IF ABS(T-T2)>2 THEN PRINT
    "NI SIQUIERA CERCA"
```

3. La línea 170 contiene el número que determina la longitud del tiempo de vida. Cambia el 50 por un número más alto para mayor tiempo de vida.

Página 23
El valle de la Muerte

Puedes hacer el valle más largo cambiando el número en la línea 30 por otro mayor que 200.

Página 25
Minas espaciales

Añade estas líneas para hacer que el ordenador te pregunte si quieres jugar de nuevo:

```
645 PRINT "OTRO JUEGO? (TECLEA S/N)"
646 INPUT A$
647 IF A$="S" THEN GOTO 10
```

Debes entonces poner una nueva línea 5 y cambiar la 30 para añadir el dinero que tienes al final del juego al dinero permitido al principio del nuevo juego:

```
5 LET M=0
30 LET M=M+INT(RND*50+10)*P
```

(Asegúrate de usar la versión correcta de RND para tu ordenador.)

Título original: COMPUTER SPACEGAMES

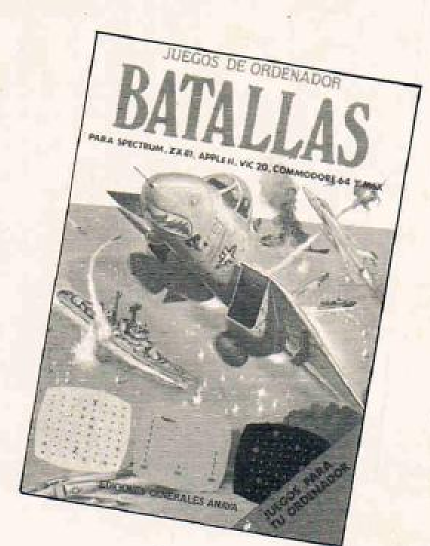
Villafranca, 22. 28020 Madrid
I.S.B.N.: 84-7525-313-X
Depósito legal: M-39.269/1985
Impreso por: Melsa
Carretera de Fuenlabrada
a Pinto, km 21.800
Pinto (Madrid)
Impreso en España - Printed in Spain

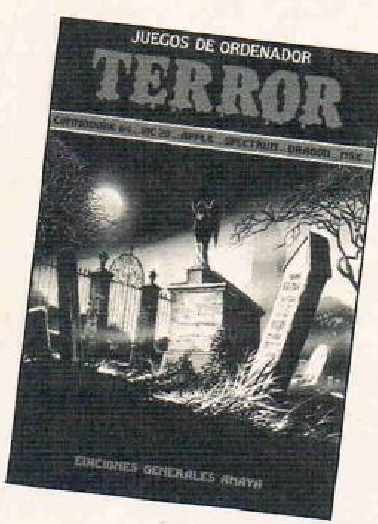
JUEGOS DE ORDENADOR

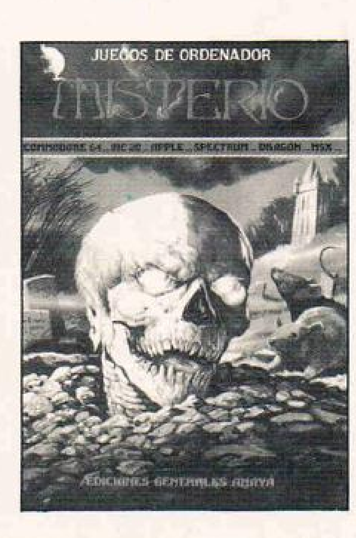
Cada uno de estos nuevos libros contiene sencillos programas de juegos para utilizar en los ordenadores domésticos de mayor difusión. Junto a cada programa hay explicaciones sobre cómo funciona, así como problemas y sugerencias para introducir cambios que compliquen las aventuras o las hagan más sencillas, según los deseos del usuario. Por medio de estos apasionantes juegos, incluso los que nunca hayan utilizado un ordenador aprenderán con facilidad cómo funciona un programa sencillo, y podrán diseñar alguno por sí mismos. Al final de cada libro aparecen las claves de conversión para diversos ordenadores, así como un glosario de los términos más usuales en BASIC.

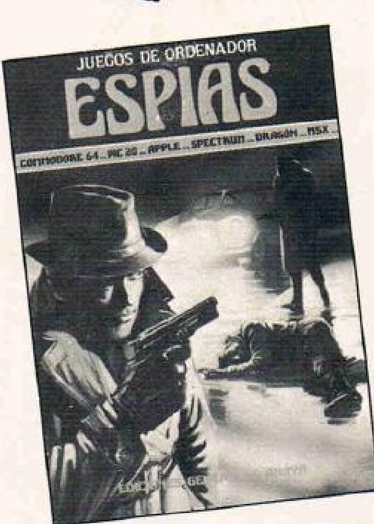
Títulos de la serie











EDICIONES GENERALES ANAYA